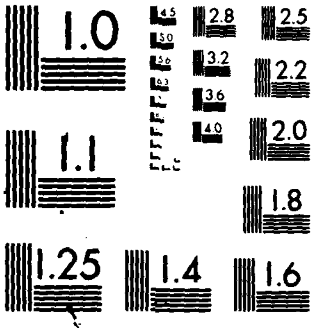
AD-A098 783   THRESHOLD TECHNOLOGY INC DELRAN NJ   F/G 9/2
LIMITED CONNECTED SPEECH T&E.(U)
MAR 81 E SHAMSI, J R WELCH   F30602-79-C-0240

UNCLASSIFIED   RADC-TR-81-21   NL

1 of 1
AD
A098783

END
DATE
FILMED
8-81
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

12

RADC-TR-81-21
Final Technical Report
March 1981

LIMITED CONNECTED SPEECH T&E

Threshold Technology, Inc.

Edward Shamai
John R. Welch

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, New York 13441

This report has been reviewed by the RADC Public Affairs Office [illegible] is releasable to the National Technical Information Service [illegible] it will be releasable to the general public, including foreign nations.

RADC-TR-81-21 has been reviewed and is approved for publication.

APPROVED: John V Ferrante

JOHN V. FERRANTE, 1/Lt, USAF
Project Engineer

APPROVED:

OWEN R. LAWTER, Colonel, USAF
Chief, Intelligence and Reconnaissance Division

FOR THE COMMANDER: John P. Huss

JOHN P. HUSS
Acting Chief, Plans Office

SUBJECT TO EXPORT CONTROL LAWS

If your address has changed or if you wish to be removed from the RADC mailing list, or if the addressee is no longer employed by your organization, please notify RADC (IRAA) Griffiss AFB NY 13441. This will assist us in maintaining a current mailing list.

Do not return this copy. Retain or destroy.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

19 REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
18 RADC-TR-81-21

2. GOVT ACCESSION NO.
AD-A098 783

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
6 LIMITED CONNECTED SPEECH T&E

5. TYPE OF REPORT & PERIOD COVERED
9 Final Technical Report

6. PERFORMING O7G. REPORT NUMBER
N/A

7. AUTHOR(s)
10 Edward Shamsi
John R. Welch

8. CONTRACT OR GRANT NUMBER(s)
15 F30602-79-C-0240

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Threshold Technology, Inc.
1829 Underwood Blvd
Delran NJ 08075

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
62702F
16 45941577 17 15

11. CONTROLLING OFFICE NAME AND ADDRESS
Rome Air Development Center (IRAA)
Griffiss AFB NY 13441

12. REPORT DATE
11 Mar [redacted] 81

13. NUMBER OF PAGES
28 12

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)
Same

15. SECURITY CLASS. (of this report)
UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Same

18. SUPPLEMENTARY NOTES

RADC Project Engineer: John V. Ferrante, 1/Lt, USAF (IRAA)

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Connected Word Recognition
Connected Speech Recognition
Speech Recognition
Acoustic Phonetics

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report describes the configuration and testing of a connected word recognition system. This system is capable of recognizing a total vocabulary of 80 words partitioned into 10 nodes, with maximum of 17 words active in each node. Sixty-four words of this vocabulary can be recognized in connected fashion, with a maximum of 10 words in a string.

For this system, an average recognition accuracy of 94.1 percent was achieved by experienced speakers.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

# TECHNICAL REPORT SUMMARY
for
LIMITED CONNECTED SPEECH T&E

1. Technical Problem

The objective of this program has been to develop an automatic speech recognition system capable of inputting connected speech into a computer in real-time using an 80-word vocabulary. Training of the system was to be speaker-dependent and under control of the user.

2. General Methodology

In order to use the time during which the speech data is being entered, the system is organized to operate on a sample-by-sample basis. Hence, the recognition results can, in principle, appear immediately upon detection of the end of the speech string. The system consists of a preprocessor, sample averager, a time sample correlator, a word match processor, and a string match processor. Except for the preprocessor all these subsystems are contained in the central processor.

In this system, we have used as input data the same basic feature set and word boundary circuits that have been successfully employed in Threshold's word recognition systems.

The preprocess samples and provides these features to the CPU every 2.2 msec to detect beginning of string and end of string boundaries. After the beginning of the string has been detected, the feature vectors are averaged in successive groups of seven vectors to produce a threshold average vector at a constant 15.4 msec sampling interval. The length of the string is constrained to lie between 40 samples (the minimum string length for a 10-word string) and 500 samples (the maximum length for a 10-word string).

As each 32-bit time sample is generated at 15.4 msec intervals, it is correlated against all time sample vectors for all of the words in the connected word vocabulary and the results are stored in a correlation result buffer. This buffer is large enough to hold the time sample correlation data for matching all of the reference arrays against 50 input samples, corresponding to the longest possible word.

The next stage of processing uses the correlation results to perform a linear-path word matching between the input time samples and all of the words in the vocabulary. For each possible word end point, this match is performed for all starting points within the constaints of the assumed minimum and maximum word lengths. When this word matching process is completed for a particular word end point, it generates an array of "best" word match results and "best" word match scores for all possible word starting points corresonding to that end point.

The final process in the system is the dynamic programming string matching algorithm. This algorithm takes the start point-end point score data and performs an efficient search to find the combination of start and end points which gives the best string score from the beginning of the string up to that word end point for all possible numbers of words that can fit within the given number of speech samples. As this final string match proceeds, it generates a pointer array so that the string which provides the best match can be reconstructured. The final string match is made by choosing the overall best string between the start and end points, or if the number of words per string is prespecified, the best string with that number of words.

From an operational point of view, this system can recognize a maximum of 80 vocabulary words partitioned into a maximum of 10 nodes with a maximum of 17 words active in each node. The vocabulary words can functionally be broken into three sets, namely, node names, command words, and regular vocabulary words to be recognized in connected fashion.

3. Summary of Accuracy Tests

Training and testing was done by two groups of speakers. The first group of speakers was largely unfamiliar with voice data entry systems, while the second group of speakers was all familiar with such systems.

With the first group of 55 speakers, (39 males and 16 females), an average recognition accuracy of 90.68 percent was obtained. The second group, consisting of 9 speakers (7 males and 2 females), achieved an average recognition accuracy of 94.1 percent. Over all tests, the highest speaker recognition accuracy achieved was 99.3 percent.

It should be mentioned that the test data was recorded three months after the training data was recorded.

TABLE OF CONTENTS

APPENDICES

LIST OF ILLUSTRATIONS

Figure

Table

EVALUATION

This effort is part of the Center Program being conducted under Project 4594 to provide improved data entry capabilities required for use with today's high speed processors. The effort was initiated to develop a more natural version of Voice Data Entry. Present systems are isolated word systems which, although highly reliable, are an unnatural form of communication for operators.

Under this effort, algorithms were investigated to develop a connected speech recognition system. The system developed was capable of handling an 80-word vocabulary with 10 nodes or subsets of the vocabulary with a maximum of 17 words per node. A maximum of 10 words is allowed in each entry sequence spoken in connected fashion. Tests were conducted using 3 digit strings and variable length strings of 3 to 7 words. An average accuracy of 90.5% was achieved in these tests. No A-Priori knowledge was assumed by the contractor in conducting these tests. The addition of syntax rules, such as the # of words in the sequence or knowledge of the makeup of the input string would improve the accuracy of the system.

The algorithms developed during this effort are being installed in a system at RADC and further tests will be run. More work in this area is planned especially in the area of coarticulation between adjacent words.

John V. Ferrante

JOHN V. FERRANTE, 1/Lt, USAF
Project Engineer

INTRODUCTION

A connected word recognition system has been developed which is an interim step between presently available isolated word recognition systems and speech understanding systems. This report is provided to explain the operation and performance of this system.

Providing a fast and easy-to-use data entry system is a goal of scientists in the field of man-machine communications, which has been partially achieved by presently available isolated word recognition systems. In order for a simple isolated word recognition system to operate successfully, however, users have to leave pauses of 100 to 200 msec duration between words. This requirement limits the use of these systems to slow, single-word-at-a-time data entry.

Recently, a more sophisticated isolated word recognition technique has been devised by Threshold Technology Inc. (Threshold) called QUIKTALK^{TM}, in which silence gaps between words can be reduced to as little as 20 msec. This system allows much greater data entry speed than a simple isolated word recognition system, but since it is still necessary to leave some silence between words, effective use of the system still requires an unnaturally choppy speaking style. Consequently, it is desirable to provide a speech recognition system which eliminates any need for leaving pauses between words. Such systems are said to provide "connected word" recognition.

Simple "connected word" recognition systems should be differentiated from "continuous speech" recognition systems which are also sometimes referred to as "speech understanding" systems. In connected word recognition systems, the vocabularies are limited and word reference templates are generally obtained from isolated training utterances. Hence, there is a limit to the amount of running together of words or co-articulation that can be tolerated in the recognition mode. In speech understanding systems, however, the goals may be to allow the speaker to introduce nearly as much word distortion or co-articulation as would occur in normal conversation.

Although continuous speech recognition may be an ideal means of communication between man and machine, the work in this area is still in the early experimental stages, and many years will pass before such systems will be available for practical applications. In the meantime, therefore, there is a need for voice input systems which are both faster and more natural to use than most presently available isolated word recognition systems. To provide for this need, at a reasonable cost, is the objective of the work that is reported in this document. Section II of this report describes a connected word recognition system which in many respects has been designed to be an extension of a very successful existing isolated word recognition system.

The basic approach is to perform connected word recognition without preliminary segmentation of the speech signal into words. The way that this is done, of necessity, requires the testing of many different alternative word start and end points, and is computationally demanding. Because of the heavy computational requirements, much of Section II is a discussion of techniques for reducing the computational burdens.

Section III is a description of the response of the system to an extensive series of tests and a discussion of some of the deficiencies of the system with suggestions for improving system performance.

HOW THE CONNECTED WORD RECOGNITION SYSTEM IS ORGANIZED

In order to use the time during which the speech data is being entered, the system is organized to operate on a sample-by-sample basis. Hence, the recognition results can, in principle, appear immediately upon detection of the end of the speech string. The system consists of a preprocessor, sample averager, a time sample correlator, a word match processor, and a string match processor.

The connected word recognition system operates as shown in Figure 2.1. The speech is broken down into 32 binary features by the Threshold Technology Preprocessor. These features are an amplitude-normalized set of spectral slopes, spectral maxima, and phoneme class features which have been selected and proven to be effective for discriminating spoken words. These features are sampled every 2.2 msec by the preprocessor to detect beginning of string and end of string boundaries. After the beginning of the string has been detected, the feature vectors are averaged by the CPU in successive groups of seven vectors to produce a threshold average vector at a constant 15.4 msec sampling interval. The length of the stream will be constrained to lie between 40 samples (the minimum string length for a 10-word string) and 500 samples (the maximum string length for a 10-word string).

As each 32-bit averaged vector is generated at 15.4 msec intervals, it is correlated against the reference patterns for each word active at the time. A reference pattern for a word consists of 16 time samples of a 32-bit vector, thus the incoming averaged vector is correlated against each time sample. The results are stored in a circular correlation result buffer. This buffer is large enough to hold the time sample correlation data for matching all of the reference arrays against 50 input average vectors, corresponding to the largest possible word.

The next stage of processing uses the correlation results to perform a linear-path word matching between the input time samples and all of the words in the vocabulary. For each possible word end point, this match is performed for all starting points within the constraints of the assumed minimum and maximum word lengths. When this word matching process is completed for a particular word end point, it generates an array of "best" word match results and "best" word match scores for all possible word starting points corresponding to that end point. To reduce computation only every other sample (30.8 msec) is assumed to be a valid start point. Also each start point is assumed to be the end point of the previous word.

The final process in the system is the dynamic programming string matching algorithm. This algorithm takes the start point-end point score data and performs an efficient search to find the combination of start and end points which gives the best string score from the beginning of the string up to that word end point for all the possible numbers of words that can fit within the given number of speech samples. As this final string match proceeds, it generates a pointer array so that the string which provides the best match can be reconstructed. The final string match is made by choosing the overall best string between the start and end points, or if the number of words per string is prespecified, the best string with that number of words.

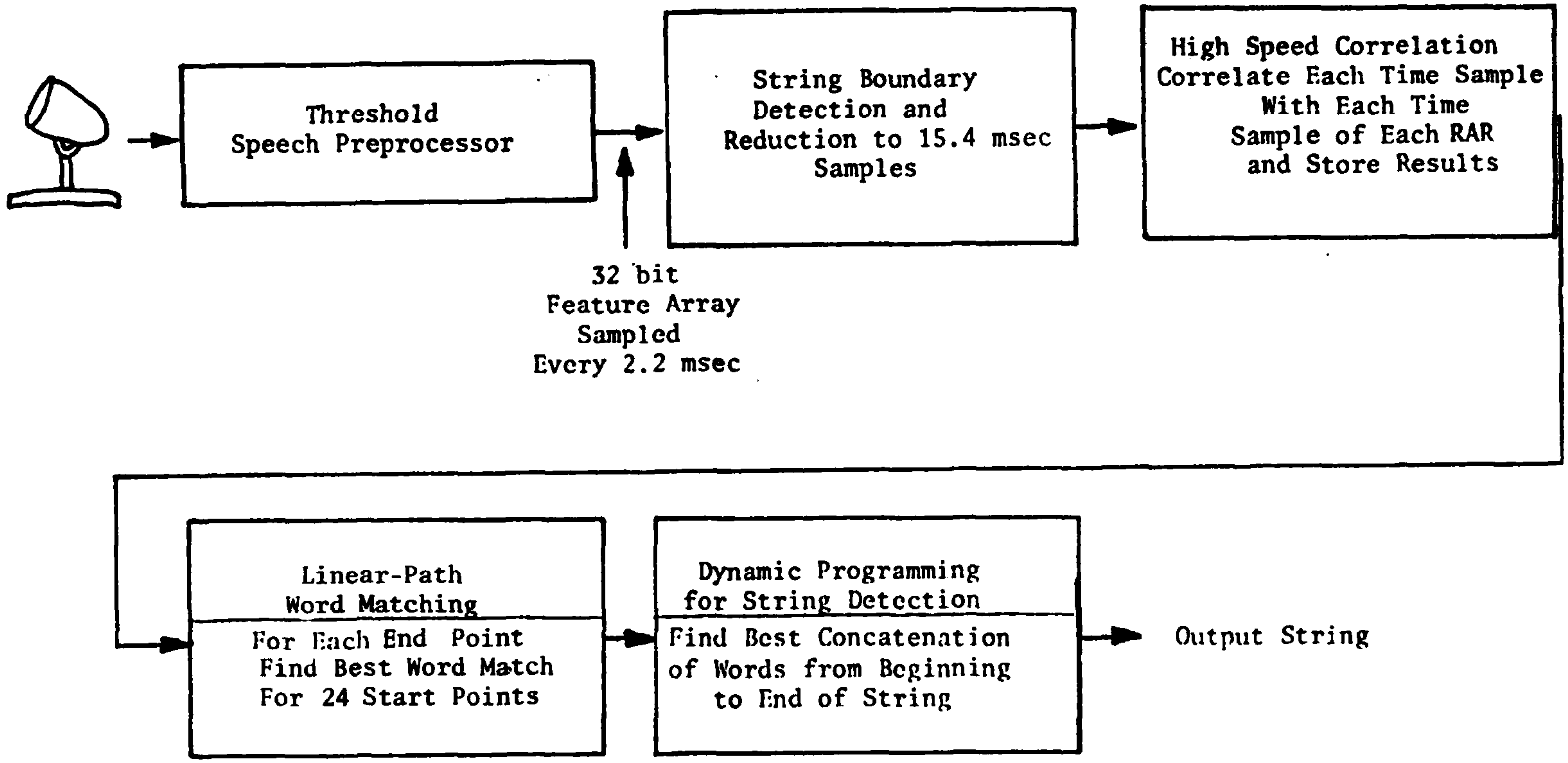


Figure 2.1 Connected Word Recognition System

NODE STRUCTURE AND COMMAND WORDS

A capability of partitioning a maximum of 80 words into a maximum of 10 usable nodes is provided. Six command words are available for system control functions such as activating a training mode, etc.

A maximum of 12 nodes exist in this system (however, two are system nodes not available to user). The first node, which is referred to as the base node (or node Ø), contains all of the node names and commands available to the system. The next 10 nodes are nodes which actually partition the vocabulary words. The last node idles the system so that subsequent conversation will not be recognized.

This system is capable of recognizing a total vocabulary of 80 words. The vocabulary is functionally divided into three major subsets. One subset includes 10 node names, each assigned to nodes 1 through 10. These nodes can be selected by the user by speaking the corresponding node names. The other subset contains a maximum of 64 vocabulary words to be chosen by the user. The last subset includes 6 command and control words to be spoken and recognized as isolated words. These words and the functions which they perform are presented below.

GO: By uttering the word "GO" in isolation from any given node, the user can enter into node Ø.

CANCEL: Upon recognition of this word in isolation, the last line outputted to the CRT will be erased.

RETRAIN: By uttering the word "RETRAIN" in isolation while node Ø is active, the training mode of the system will be activated.

OFFLINE: By uttering this word twice in isolation while node Ø is active, the system will idle, so that subsequent conversation will not be recognized.

RESTART: By uttering the word "RESTART" twice in isolation, an operator can exit the offline node and enter into node Ø.

TUNE-UP: This command allows an operator to evaluate the effectiveness of his training data.

Upon recognition of this word in isolation, the first word active in the current node will appear on the CRT. At this time, the operator should utter this vocabulary word. If the SCORE obtained for this word passes a certain threshold, the next word in the current node will appear on the CRT. Otherwise, the same word is requested to be uttered once more. If the obtained score passes the threshold, the next word in the current node will appear on the CRT. Otherwise, the word number, obtained score, and the threshold from the rejected word will be stored in a buffer, so that later these data can be outputted at the end of the tune-up operation.

If an operator wishes not to perform the tune-up operation for a word which has appeared on the CRT, he can hit the carriage return key (CR) so that the next word in the node will be prompted.

The Connected Word Recognition System

USE OF THE THRESHOLD PREPROCESSOR FOR FEATURE EXTRACTION AND PAUSE DETECTION

In the present connected word recognition system, we have used as input data the same basic feature set and word boundary circuits that have been successfully employed in Threshold's word recognition systems.

The present Threshold connected word recognition algorithm uses, as input speech features, the 32 outputs of the Threshold 8040 preprocessor.

For connected speech recognition, accurate boundary detection is necessary to define the true beginning and end of strings of connected words. Threshold employs the same sophisticated pattern recognition techniques to determine string boundaries as have been successfully used in Threshold's isolated word recognition systems to determine word boundaries. A hierarchy of features is measured and thresholds are set to distinguish vocabulary words from background noise and extraneous non-speech utterances such as coughs, sneezes, lip smacking, and breathing noises.

The Connected Word Recognition System

CONNECTED WORD RECOGNITION WITHOUT PRELIMINARY SEGMENTATION INTO WORDS

Reliable connected word recognition can be achieved without preliminary segmentation into words by simultaneously finding the best string of words and the best choice of word boundaries for fitting those words between the first and the last speech samples in the string. This is done by assuming many possible start and end points for each word in the string, and by matching all of the words in the vocabulary with the speech data between each assumed pair of start and end points.

Because of the demonstrated low reliability of most available word, syllable, and phoneme segmentation techniques, we accomplish connected word recognition without relying on preliminary segmentation data. Instead, word recognition and segmentation is achieved by direct pattern matching between feature reference templates obtained for each word during a training phase and the input data string to be recognized. Hence, the correlation results themselves provide segmentation information. In order for this to work, however, it is necessary to match the concatenated words in the input data against many possible strings of concatenated words of reference data. The match is made over many possible start and end points in the data string, and over all of the words in the vocabulary between each assumed pair of start and end points. This technique is, in effect, a direct extension of the technique which is successfully employed in the Threshold isolated word recognition systems, in that no apriori phonetic assumptions are made. Instead, all recognition is done by matching input data to data which has been extracted during training. In addition, all of the demonstrated word recognition power of the Threshold recognition features and pattern matching methodology is applied directly to the problem.

On first appearance, the number of possible combinations of word start and end points and the number of word match correlations appears to be so great as to preclude a practical implementaion of this technique. In the following sections, however, we will show that by systematic organization and, in particular, by using dynamic programming to reduce the complexity of the string match and word match problem, this technique can be implemented in real-time.

CHOOSING LINEAR TIME NORMALIZATION FOR GENERATING THE WORD REFERENCE DATA

Linear time normalization of training repetitions has been adopted over non-linear time normalization because it provides a variable time base for accommodating different speaking speeds, eliminates dependence upon any one single repetition, and simplifies the necessary matching algorithm.

The connected word recognition system uses the same training method employed in the Threshold isolated word recognition algorithms for training the words. Each vocabulary word is spoken five times. Each individual repetition is linearly divided into sixteen equal time slots. The five repetitions are then averaged, on a time slot basis, to form one reference array of sixteen time slots to represent that vocabulary word.

An alternative method is non-linear time normalization with dynamic programming. Each vocabulary word is spoken five times and the average length of these repetitions is found. Each individual repetition is optimally warped to the one sample whose length is closest to the average using dynamic programming. The five repetitions are then averaged on a sample basis to form one reference array to represent that vocabulary word. The length of each reference array is the length of the array that was used as a time warping reference during training.

There are several advantages to using linear time normalization to a fixed number of reference time slots. Normally, there are substantial differences between speaking rates during training and testing, and this procedure will automatically adjust to those differences. In addition, by normalizing all repetitions of a given word to the same length, the averaging can be done without choosing one sample as an initial time reference.

A disadvantage of using the same number of time slots for each word is that the duration information is lost. We have found, though, that absolute duration information during training is not meaningful because users generally speak differently during training than during operation. Relative duration data, however, does have some significance during training.

IMPLEMENTATION OF LINEAR TIME NORMALIZATION DURING TRAINING

Linear time normalization of training repetitions has been implemented with a simple algorithm which uses a minimum amount of memory. Word duration information, present during system training, has not been utilized in the present connected word recognition system.

In the present system, speech input is sampled at 2.2msec time intervals. Each sample contains thirty-two bits which represent thirty-two characteristic features which are derived in the preprocessor. A logic 1(0) means that the feature is on(off) at that time sample. These thirty-two bit samples are temporarily stored in sixteen-bit computer word pairs in an input buffer array.

Each vocabulary word is spoken five (or ten) times. Each repetition consists of approximately 90-260 samples which are linearly divided into sixteen equal time slots. The samples in each time slot are combined to form one representative sample for that time slot. A particular feature is considered on (off) in the representative sample if it is on (off) for more than 1/4 (3/4) of the time slot. The result of this step is sixteen representative samples for each repetition of the vocabulary word. Refer to Figure 2.2.

These repetitions are then averaged on a sixteen time slot basis to form one reference array (RAR) for that particular vocabulary word. Each RAR consists of two arrays referred to as the most significant bit (MSB) and the non-extremum bit (NEB). The MSB indicates whether a certain feature has occurred and the NEB indicates the frequency of occurrence. If a feature in a time slot is either off or on virtually all of the time, then the non-extremum bit is set to 0, otherwise, it is set to 1. This array increases resolution and, ultimately, recognition accuracy. 64 computer words of memory are required for each spoken word.

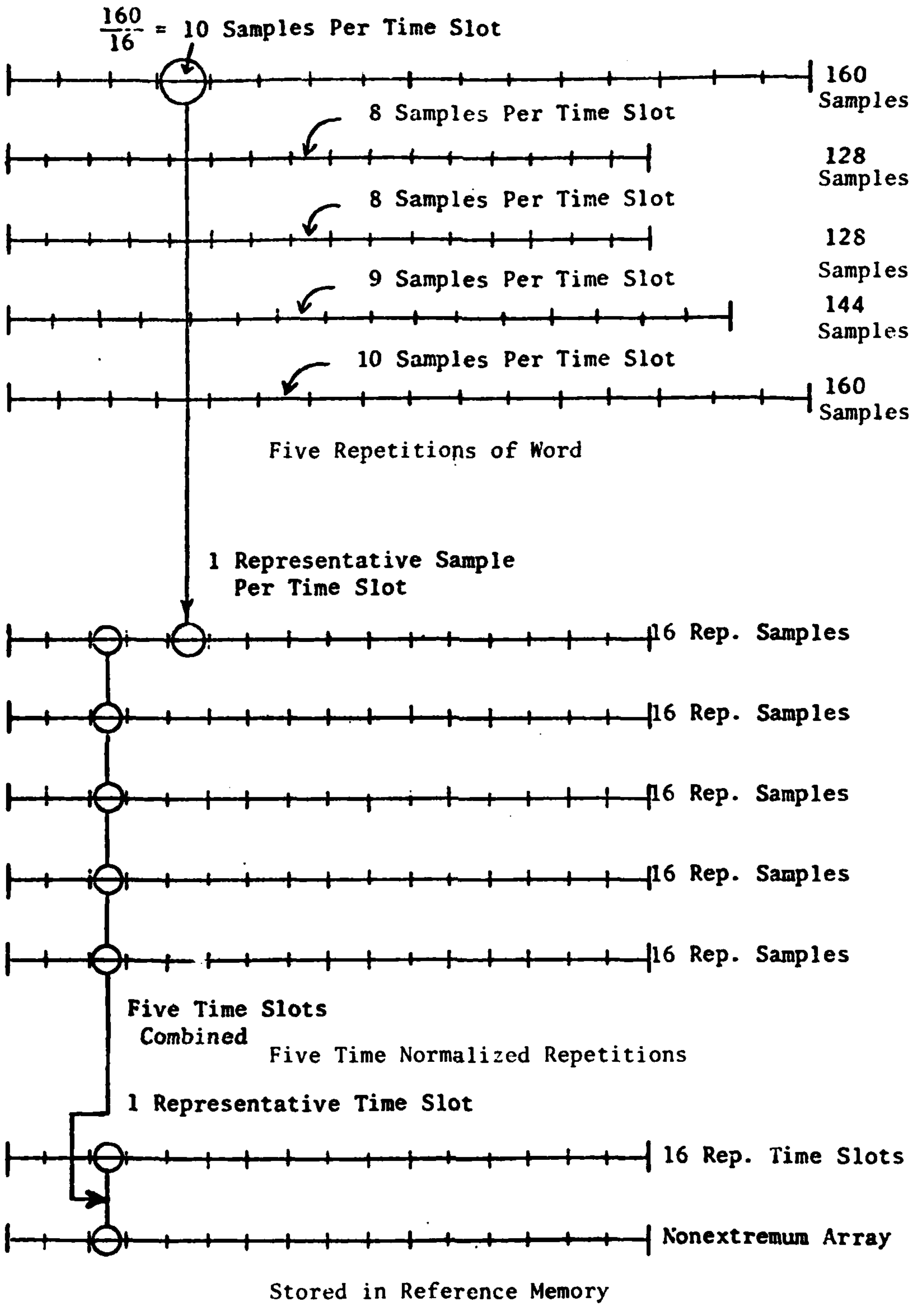


Figure 2.2 Linear Time Normalization Shown for Five Repetitions

UPDATING THE TRAINING DATA

An updating routine has been provided to allow updating of vocabulary words with only two training repetitions per word.

In the present system, the reference array (RAR) consists of two arrays, referred to as MSBs and NEBs, which are generated during training. These two arrays provide information about the presence or absence of a feature and the consistency of its occurrence, respectively. The MSB (Most Significant Bit) is set if the feature tends to be on more than off in a number of training repetitions, and the NEB (Non-Extremum Bit) is set if the feature is not extremely consistent in the training repetitions. With time, the speaker's voice can change and therefore, it would be appropriate to allow the user to update the RARs to maintain the recognition accuracy.

An updating routine has been provided to allow updating of vocabulary words with only two repetitions per word. Table 2.1 provides information about how updating is performed. Since updating consists of two utterance for each word, each of the thirty-two features of a word could assume the following set of conditions (00, 01, 10, 11). For example, the condition 11 indicates that a given feature was on for both utterances and 1Ø indicates that a given feature was on in the first utterance and off for the second one. These possibilities are listed in the column titled FARs. The next column indicates the possible ways which MSB and NEB bits for a given feature might be set. The last column shows the new MSB and NEB. This column is set according to the conditions of the old MSB and NEB and the information provided in the FARs column.

UPDATING PROCEDURE

	OLD		NEW	
FARS	MSB	NEB	MSB	NEB
00	1	0	1	1
	1	1	0	1
	0	1	0	0
	0	0	0	0
01	1	0	1	1
	1	1	1	1
	0	1	0	1
	0	0	0	0
10	1	0	1	1
	1	1	1	1
	0	1	0	1
	0	0	0	0
11	1	0	1	0
	1	1	1	0
	0	1	1	1
	0	0	0	1

TABLE 2.1

UPDATING PROCEDURE

LINEAR-PATH WORD MATCHING

Linear-path word matching is the fastest way to match time samples of the input data with the time samples of the stored reference arrays. For efficiency, every other sample is assumed to be a possible start and end point. Each repetition of the linear-path matching computes the best word score between one end point and each possible start point corresponding to that start point. This multiple matching process is then repeated for all possible start points in the input data.

For each possible pair of starting and ending points within the string of connected words, the word matching algorithm will determine which of the active vocabulary words best matches that part of the string.

Figure 2.3 illustrates the linear-path word matching process. In this figure, the reference sample R(i) is plotted along the ordinate, and the variable test array samples T(j) are plotted along the abscissa

For an assumed end of word point, E, the possible starting points lie along the abscissa and are bounded by paths, EL, corresponding to a minimum allowed word length of 61.6 msec, and EU, corresponding to maximum allowed word length of 770 msec.

In order to provide faster response, only every other point is taken as a possible starting point. Since it is assumed that the first sample of one word is immediately preceded by the last sample of the previous word, each starting point is also an ending point. Taking every other sample as potential starting and ending points has been found to provide adequate end point resolution and reduces the word matching processing time by a factor of four.

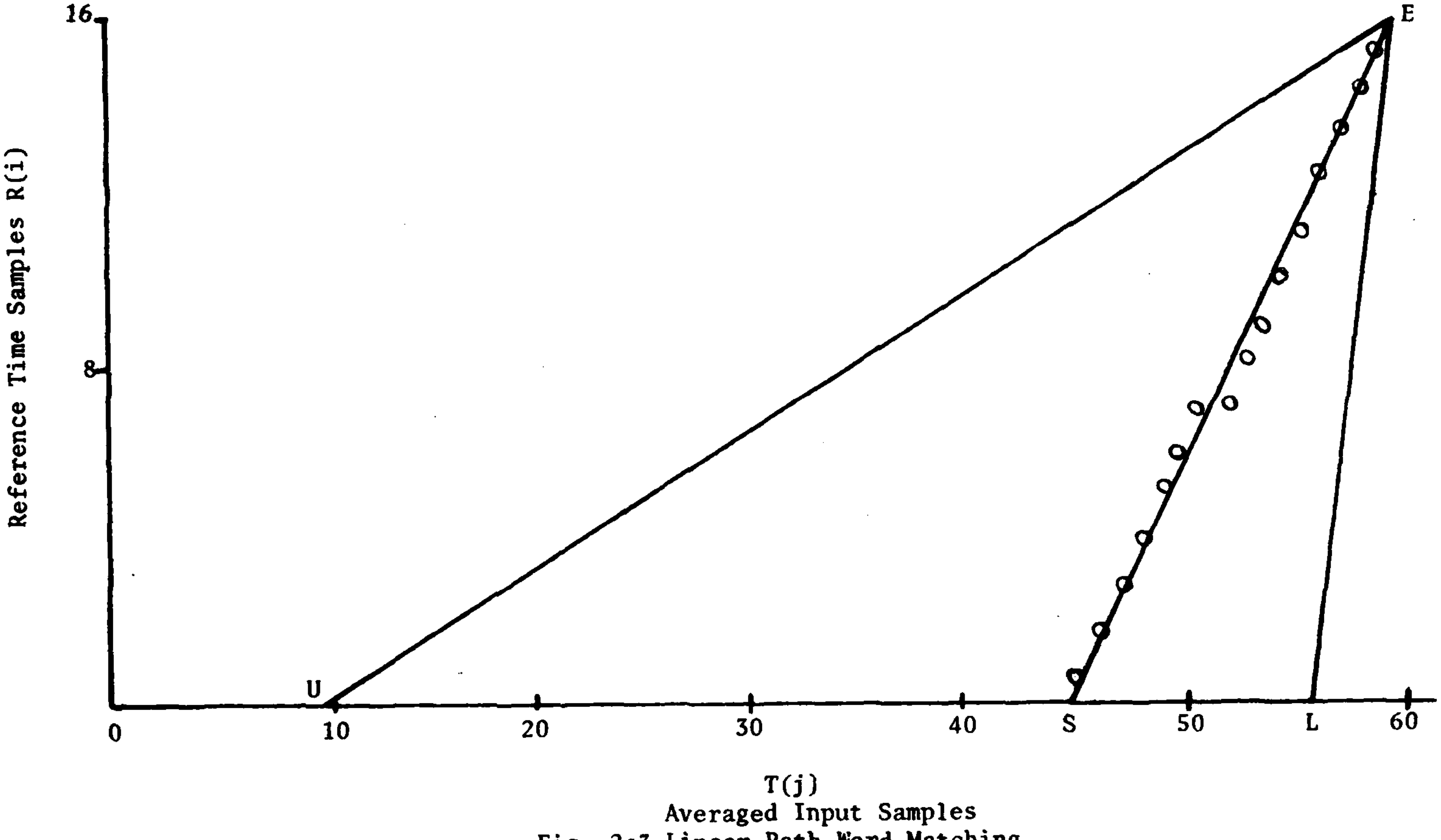


Fig. 2•3 Linear-Path Word Matching

The Connected Word Recognition System

LINEAR-PATH WORD MATCHING EQUATIONS

The problem is to find a linear time warping function for matching reference R(i) with input data T(j). This warping function is presented below.

In order to solve the problem of linear-path word matching, we have to define a linear time warping function for matching reference R(i) for i=1, I with the input data T(j) for j=1, J for a fixed ending point and for all possible starting points.

For notational purposes, we define the warping function with respect to a third index, k, to map the points of the test data to the points of the reference data. Such a function is given by Equation 1 and is simply a sequence of i and j index values with boundary conditions given by Equation 2. Equations 3 and 4, in which n denotes the slope of the path, are used to generate linear time warpoing functions for different paths. Equation 3 finds the corresponding j coordinate for a given i coordinate and Equation 4 does the reverse. These two separate equations are used so that each input sample could be matched against at least one time slot of a reference array and each time slot of a reference array could be matched against at least one input sample. Warping function for path ES is shown in Figure 2.3 by circular points along this path.

A distance measure is defined between the reference and test time samples at each point, k, of the warping function by Equation 5. In this equation, the double magnitude signs are intended to denote a general measure of dissimilarity between the indicated reference and test samples. In the Threshold systems, this distance measure is a positive-valued arithmetic complement of the correlation between the two time samples as computed by the standard Threshold correlation algorithm. The basic scoring process for a path for a given ending and starting point is represented by Equation 6. In this equation, distance values along the path are summed from the beginning to the end of the path and then the result is normalized so that the number of elements in each path is the same as the number of input samples used in the path. This computation is done along the same path for all of the active words in the vocabulary and the vocabulary word which results in the minimum (lowest) score is chosen as the word which best fits this path. This minimum value is obtained by using Equation 7, where D_m denotes the minimum distance for a given path among all the active words in the vocabulary.

This same process is repeated for all possible starting points corresponding to a given ending point and for each new ending point. The correlation points D(i,j) are computed one row at a time and stored in a rotating correlation buffer which is updated upon the receipt of each new speech sample. The results of the word matching process are passed to the string matching algorithm which searches for the minimum string distance corresponding to the string of words spoken.

$$W(k) = (i(k),j(k)) \text{ for } k = 1, K \qquad (1)$$

$$W(K) = (16,J), \text{ for } J = 6,8,\cdots, 48 \qquad (2)$$

$$j = \text{round } ((i-16) \times 1/n + 48) \quad \text{for } n \geq 1 \qquad (3)$$

$$i = \text{round } ((j-48) \times n + 16) \quad \text{for } n < 1 \qquad (4)$$

$$D(W(k)) = D(i(k),j(k)) = \| R(i(k) - T\ (j(k)) \| \qquad (5)$$

$$D_T = N(p) \sum_{k=1}^{K} D(i\ (k),j(k)) \qquad (6)$$

$$D_m = \text{Min} \left\{ D_T \right\} \text{ over all active vocabularies} \qquad (7)$$

TABLE 2.2

Linear-Path Word Match Equations

BOUNDARIES AND DIMENSIONS OF THE STRING MATCHING PROBLEM

The problem of finding the best string of vocabulary words which can be fit between the first and the last speech samples in the string, assuming all possible start and end points for each word in the string, is bounded by the assumption of minimum and maximum word lengths and by the assumption that, except for gaps, the first sample of each word is immediately preceded by the last sample of the previous word.

In order to formulate a procedure for solving the string matching problem, it is important to determine the boundaries and the dimensions of the problem. Figure 2.4 illustrates the range of possible starting and ending points which can be considered for strings of 10 or fewer words and for input speech strings of 500 or fewer samples. In order to reduce computational complexity, words are assumed to begin only at the beginnings of even numbered samples or at 30.8 msec intervals. Since the minimum word length is assumed to be four speech samples, the shortest possible strings for up to 10 words are represented by boundary AC of the diagram. Since the maximum word length is assumed to be 50 samples, the longest possible strings are represented by boundary BC.

Each horizontal line on the triangle ABC *spans* the range of points which could conceivably be the end point for the jth word, where j is the ordinate of the diagram. For example, line AB, which spans points 40 to 500, has 231 points which are the possible ending points for a 10-word string. Similarly, line DE spans the 139 points which could conceivably be the ending points for the 5th word of a string. In total, there are 1275 possible ending points in the triangle ABC. On the boundaries AC and BC, there is no flexibility to the possible word starting and ending point combinations which can be strung together to arrive at any point. Within the central regions of the triangle, ABC, however, the number of ways that each word ending point can be reached from the previous word ending point is 24, and this span is illustrated by the small triangle FGH. There is only one way to reach each of the possible 24 ending points for the first word in the string and an example is illustrated by line CI.

In general, the number of possible ways that strings can be fit between the first string sample and the jth word ending point is on the order of 24 raised to the jth power. To directly test all of these possibilities, however, is not necessary since, as will be described in the next topic, dynamic programming can be used to reduce greatly the complexity of the search for the best fitting string.

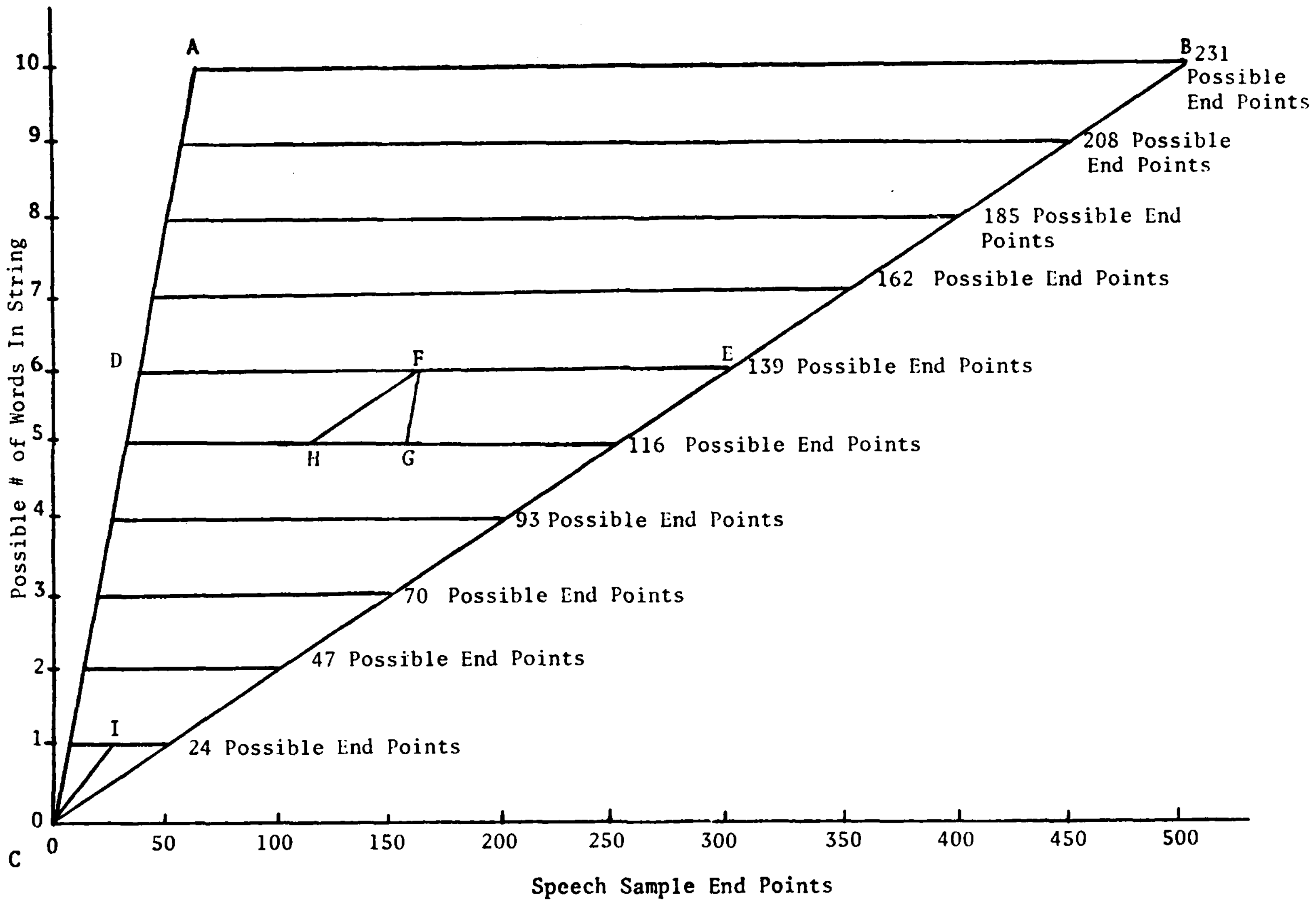


Figure 2.4 Range of String Match Possibilities

USING DYNAMIC PROGRAMMING FOR STRING MATCHING

Dynamic programming provides an efficient way to find the best string of words which can fit between the string starting and ending points. Dynamic programming is a recursive procedure which states that the best string up to a point is the one for which the sum of the best string score up to the previous ending point and best word score between that ending point and the current ending point is a minimum over all possible previous word ending points.

The dynamic programming string match procedure requires the determination of the best choice of word ending points to fit an integer number of words between the first sample and the last sample of the string. The string match procedure will first be described for the case in which the number of words is known.

String matching requires matching each word j to a word ending point e(j). In the present system, the word index will range from 1 to 10 and the possible ending points will be given by the boundaries of the string matching problem as shown in Table 2.3.

If the number of words in a string is known to be J and the last sample of the string is L, then the string matching problem is to find E(J) (the sequence of J ending points is L) as described by Equations 1 and 2, which give the minimum string score S(L,J), as described in Equation 3. In Equation 3, a string score is defined as the sum of J best word match match scores computed for J choices of ending points. Each word match score D(e(j-1)+1, e(j)) is the word match distance for the best fitting word between points e(j-1)+1 and e(j). The optimum string is defined as the string with the minimum sum of word scores, where the minimization is with respect to all possible choices of word ending points.

Direct evaluation of the string score for all possible choices of ending points would be prohibitively time consuming. Fortunately, because the string score expression is a summation, the evaluation of the optimum string can be greatly simplified by application of the recursive method of dynamic programming. In this method, optimum partial string scores are defined as in Equations 4 and 5. The optimum partial string score for the first word is simply the best string score between the first speech sample and the assumed first word ending points. This is evaluated for all possible first word ending points as described in Equation 4. For j greater than 1, the optimum partial string score for a given ending point, e(j), is computed by adding the partial string score for a previous word ending point to the best word score between that ending point and the current ending point. This sum is computed for all possible previous ending points, and the optimum partial score is the minimum over those possible ending points. A partial string score must be computed for all ending points between the lower and upper ending point limits for the word j.

In operation, word lengths will be limited to between 4 and 50 samples. Consequently, the difference between word ending points will have the limits described in Equation 6. This limitation means that Equation 5 can be rewritten as Equation 7, in which the range of the minimization search is explicitly stated.

The final string score is simply the partial string score evaluated at the end point L for word J as shown in Equation 8.

If the number of words per string is not known, apriori, the best string score is evaluated by comparing the weighted string scores for all of the possible numbers of words per string, J, and choosing the minimum over J as shown in Equation 9. The normalization factor in this equation is required to equalize the string score weights for different number of words per string prior to choosing the minimum.

TABLE 2.3

WORD ENDPOINT BOUNDARIES

Word Number	Lower Limit of e(j)	Upper Limit of e(j)
j	L(j)	U(j)
1	4	50
2	8	100
3	12	150
4	16	200
5	20	250
6	24	300
7	28	350
8	32	400
9	36	450
10	40	500

TABLE 2.4

DYNAMIC PROGRAMMING STRING MATCH EQUATIONS

(1) $E(J) = \left\{ e(1),\ e(2),\ \ldots\ldots,\ e(j),\ \ldots\ldots,\ e(J) \right\}$

(2) where $e(J) = L$

(3) $S(L,J) = \min\limits_{E(J)} \sum\limits_{j=1}^{J} D(e(j-1)+1,\ e(j))$

(4) $S_p(e(1),1) = D(1,e(1))$ for $e(1) = 4, \ldots, 50$

(5) $S_p(e(j),j) = \min\limits_{\{e(j-1)\}} \left[S_p\ (e(j-1)\ + D(e(j-1) + 1,\ e(j)) \right]$

for $e(j) = L(j), \ldots, U(j)$

(6) $4 \leq e(j) - e(j-1) \leq 50$

(7) $S_p\ (e(j),j) = \min\limits_{k=4,\ \ldots,\ 50} \left[S_p\ (e(j)-k,j-1) + D(e(j)-k+1,e(j)) \right]$

for $e(j) = L(j), \ldots, U(j)$

(8) $S(L,J) = S_p(L,J)$

(9) $S(L) = \min S(L,J)/(J)$

The Connected Word Recognition System

RESULTS

An average recognition accuracy of 94.1 percent for the connected word recognition system was achieved by experienced speakers.

The system was trained with the vocabulary of 64 words listed in Appendix B, using ten repetitions per word. While uttering each of the vocabulary words ten times, each speaker's voice was recorded on tape so that later it could be fed into the system as the training data. All of the words were trained in isolation.

The test data was recorded three months after the training data was recorded. Testing was done as follows. Each speaker uttered the scenario presented in Appendix C one line at a time. This recording was done off-line with no feedback from the system to the speaker. Later, these training and test tapes were played into the recognition system. During recognition, the vocabulary was partitioned into ten nodes with seventeen words active in each node. In the scenario of Appendix C, when each node name (spoken in isolation) was recognized, the vocabulary for that particular node was activated. Then several phrases consisting of the words active in that node were spoken in a connected fashion. Whenever the word "GO" (spoken in isolation) was recognized, the currently active node was deactivated and the first node, consisting of all node names, was activated.

Training and testing was done by two groups of speakers. The first group of speakers was largely unfamiliar with voice data entry systems, while the second group of speakers was all familiar with such systems.

With the first group of 55 speakers, (39 males and 16 females), an average recognition accuracy of 90.68 percent was obtained. The second group, consisting of 9 speakers (7 males and 2 females), achieved an average recognition accuracy of 94.1 percent. Over all tests, the highest per speaker recognition accuracy achieved was 99.3 percent.

SUGGESTIONS FOR IMPROVING PERFORMANCE OF THE PRESENT SYSTEM

By using dynamic programming word matching and providing an algorithm that can handle overlaps and gaps in speech, higher recognition accuracy can be achieved.

In the present system, a linear-path word matching algorithm is used to provide fast response time. Dynamic programming word matching with non-linear time warping results in a more flexible and therefore, more accurate word match. The tradeoff, however, is that dynamic programming requires about three times as much processing time as the linear-path word matching technique. Several techniques, however, have been devised at Threshold Technology which will reduce the processing time by a far greater factor than will be consumed by application of dynamic programming word matching.

Another problem which can be rectified is that of word overlap resulting from co-articulation. In the present system, it is assumed that the end of one word is immediately followed by the beginning of the next word in a string with no possibility for overlaps or gaps. A more flexible algorithm can be designed which takes these possibilities into account.

We have studied these problems and have devised several techniques which will be used to further improve the performance of the present system.

APPENDIX A: HOW TO ENTER THE VOCABULARY WORDS AND CONSTRUCT THE NODES

TO ENTER VOCABULARY WORDS

The maximum vocabulary size for this system is 80 words.
The vocabulary words should be entered in the following order:

1) Node Names
2) Offline
3) Go
4) Tune-up
5) Cancel
6) Retrain
7) Restart
8) Vocabulary Words

TO CONSTRUCT THE NODES

A maximum of 12 nodes exist in this system.
The first node which is referred to as the base node (or node Ø) contains all of the node names and commands available to the system. The user can enter the base node by uttering the word "GO" (in isolation) from any node of the system. Note: The word "GO" should be active in all the nodes in order to enter the base node from any given node. Nodes 1-10 are provided to partition the vocabulary words and can be activated by uttering their names in isolation while the base node is active. The last node is the "offline" node. By uttering the word "OFFLINE" twice (in isolation) while the base node is active, the offline node will be activated. This node, in effect, idles the system so that subsequent conversations will not be recognized. Uttering the word "RESTART" twice (in isolation) deactivates this node and activates the base node.

The Connected Word Recognition System

APPENDIX B: THE VOCABULARY USED FOR TESTING THE SYSTEM

Direction
Altitude
Identify
Distance
Radio
Action
Alphabet
Numbers
Maneuver
Tactics
Offline
Go
Tune-up
Cancel
Retrain
Restart
0
1
2
3
4
5
6
7
8
9
Is
Off
Yards
Feet

Bearing
Climb
Declared
Miles
Friendly
Heading
Hostile
Say
Your
What
Suspected
State
Level
ID
Initial
Frequency
Descent
Contact
Begin
Thousand
Ten
Hundred
Echo
Hotel
India
Julie
Kilo
Mother
November
Oscar

Tango
Victor
Whiskey

APPENDIX C: THE SCENARIOS USED FOR TESTING THE SYSTEM

DIRECTION

Contact Bearing 350

Contact Heading 326

State Your Heading

GO

ALTITUDE

Begin Descent 29 Thousand

Climb Level 47 Hundred

GO

IDENTIFY

Contact Suspected Friendly

What is Heading Suspected Contact

Suspected Contact Hostile

GO

DISTANCE

5302 Yards

352 Hundred Miles

527 Thousand Feet

GO

RADIO

State Your ID

Your What Frequency

GO

ACTION

State Your Contact

Begin Initial Contact

GO

ALPHABET

Mother India Echo

Julie Hotel November

Victor Whiskey Tango

GO

NUMBERS

525
202
044
843
583
171
854
349
565
113
460
964
076
737
974
357
212
453
033
248

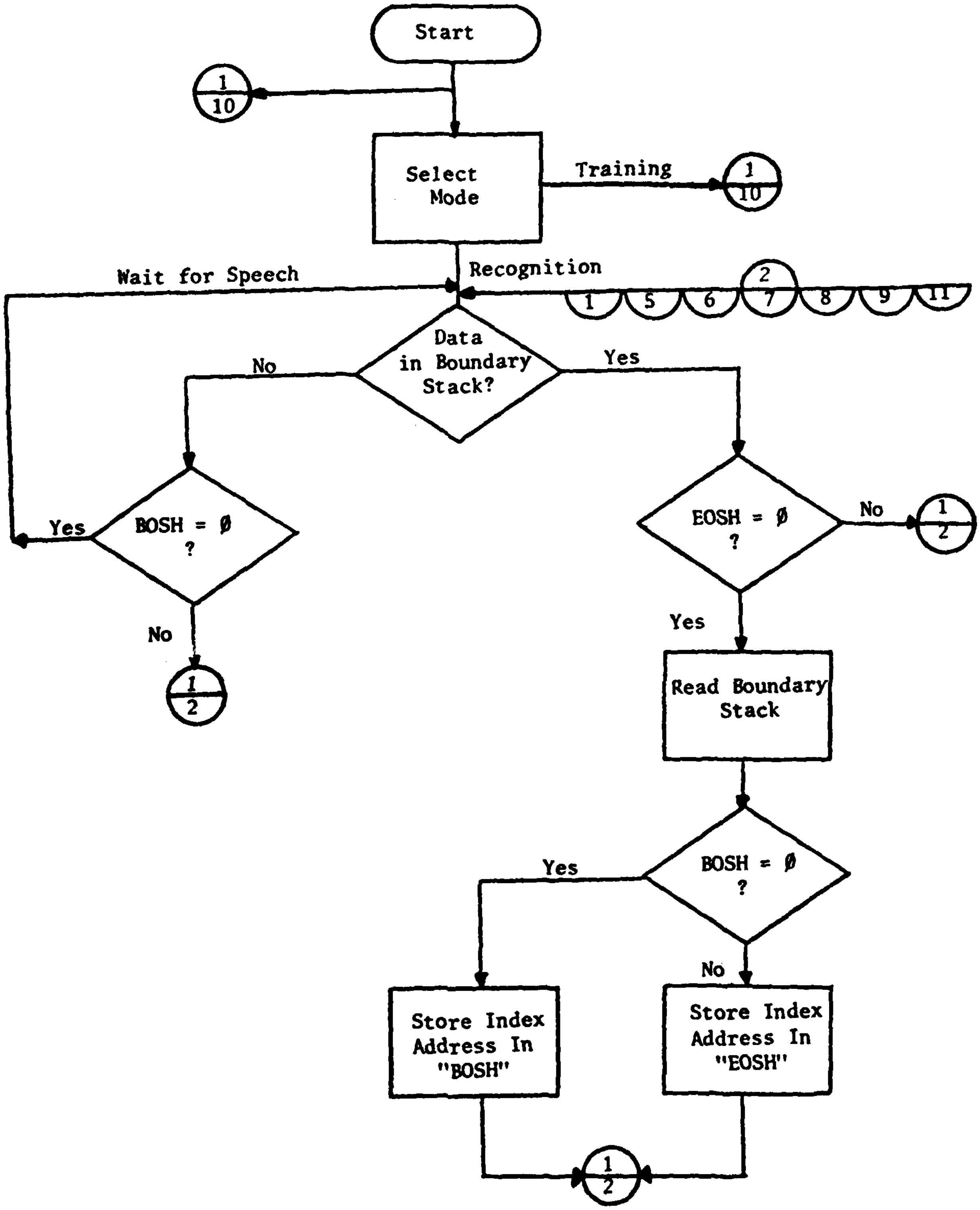
Connected Word Recognition System Flow Charts

KEY VARIABLES USED IN
THE FOLLOWING FLOW CHARTS

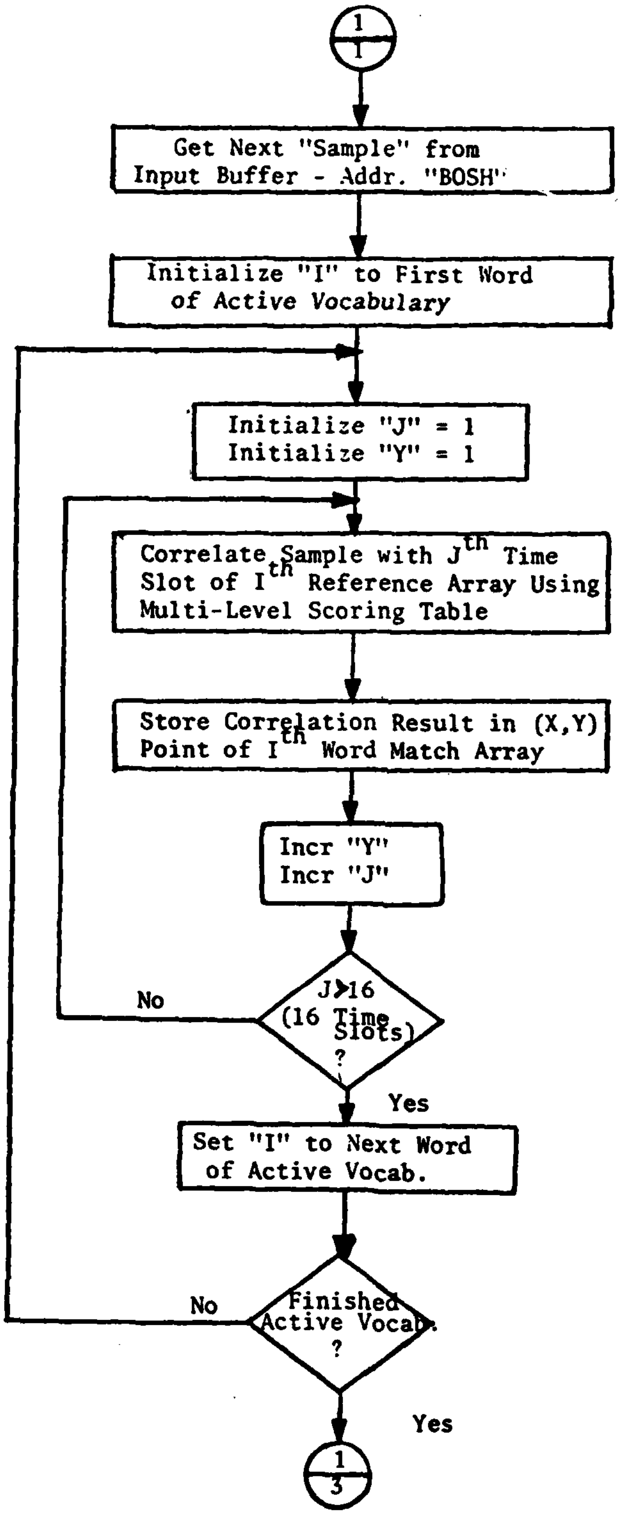
M.V.N -- Max. Vocabulary Number

M.V.S -- Max. Vocabulary Size

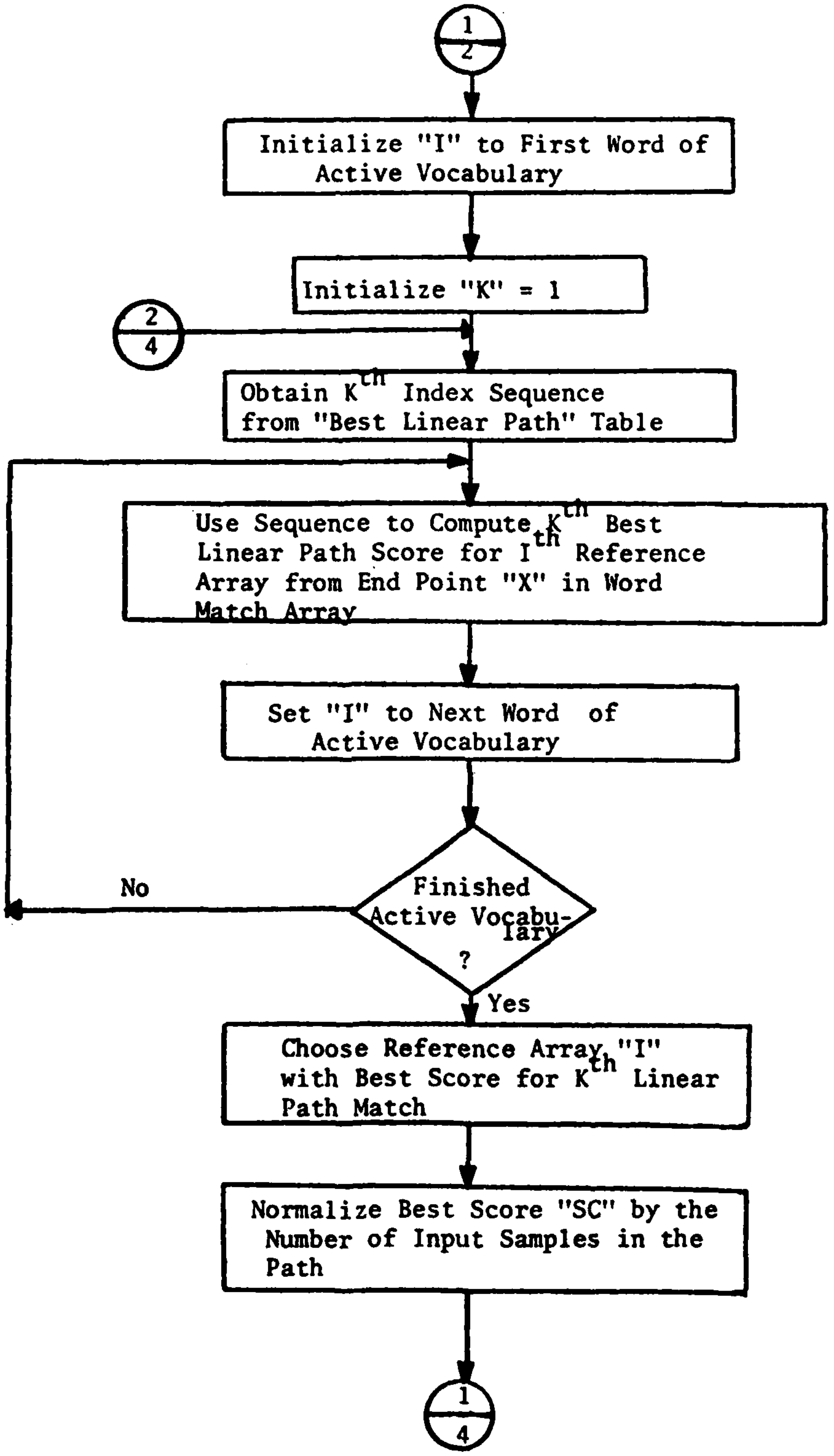
X - WORD MATCH ARRAY, INPUT SAMPLE INDEX
Y - WORD MATCH ARRAY, REFERENCE ARRAY
TIME SLOT INDEX
I - REFERENCE ARRAY POINTER
J - TIME SLOT POINTER
K - WORD MATCH ARRAY, WORD LENGTH INDEX
L - STRING MATCH ARRAY, STRING LENGTH INDEX


Start
1/10
Select Mode
Training
1/10
Recognition
Wait for Speech
2
1
5
6
7
8
9
11
Data in Boundary Stack?
No
Yes
BOSH = Ø ?
Yes
No
1/2
EOSH = Ø ?
No
1/2
Yes
Read Boundary Stack
BOSH = Ø ?
Yes
No
Store Index Address In "BOSH"
Store Index Address In "EOSH"
1/2

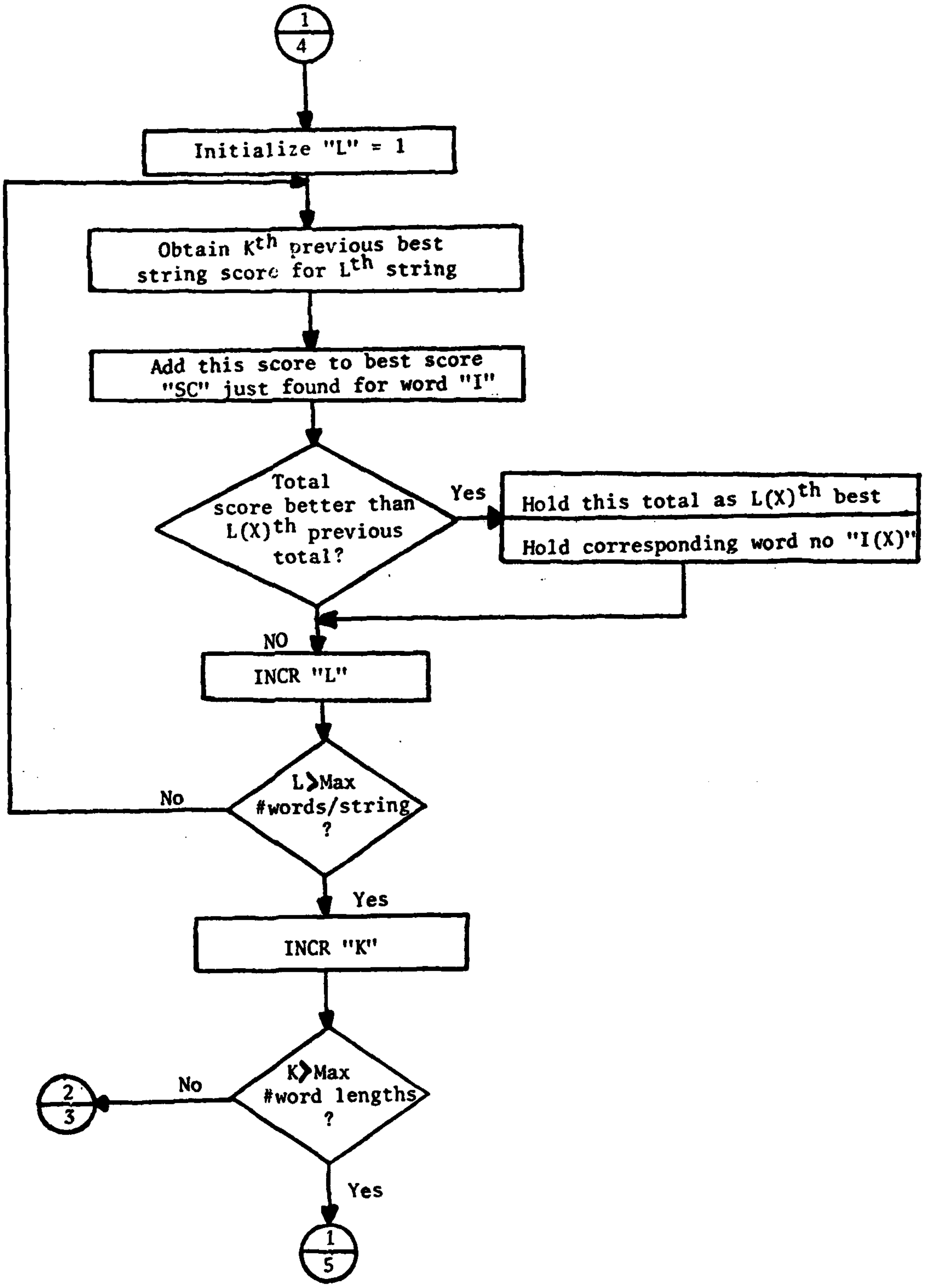
Recognition - (Correlation)


1
1
Get Next "Sample" from Input Buffer - Addr. "BOSH"
Initialize "I" to First Word of Active Vocabulary
Initialize "J" = 1
Initialize "Y" = 1
Correlate Sample with J th Time Slot of I th Reference Array Using Multi-Level Scoring Table
Store Correlation Result in (X,Y) Point of I th Word Match Array
Incr "Y"
Incr "J"
J>16 (16 Time Slots) ?
No
Yes
Set "I" to Next Word of Active Vocab.
Finished Active Vocab. ?
No
Yes
1
3

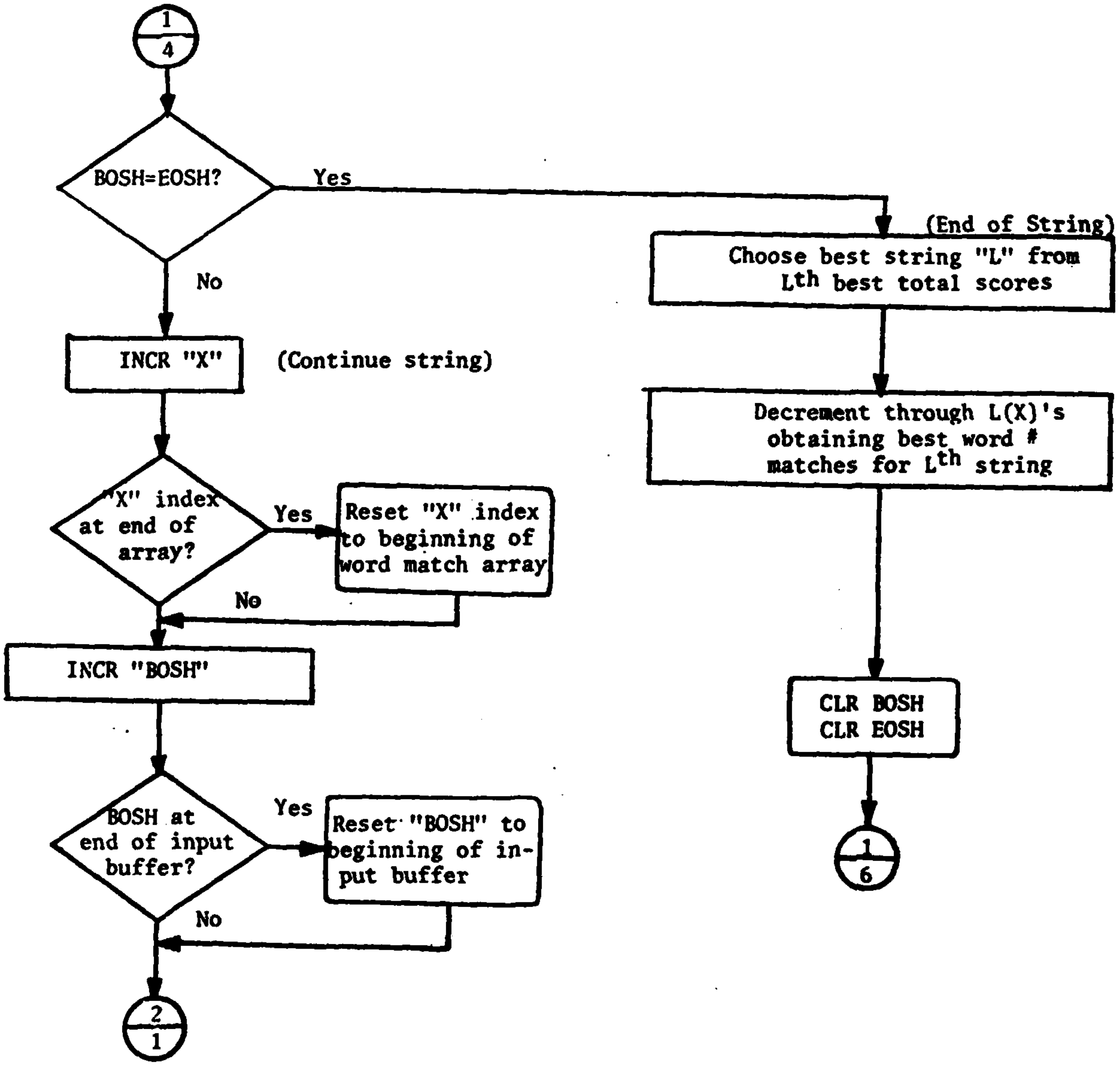
Recognition - (Word Match)


1
2
Initialize "I" to First Word of Active Vocabulary
Initialize "K" = 1
2
4
Obtain Kth Index Sequence from "Best Linear Path" Table
Use Sequence to Compute Kth Best Linear Path Score for Ith Reference Array from End Point "X" in Word Match Array
Set "I" to Next Word of Active Vocabulary
Finished Active Vocabulary ?
No
Yes
Choose Reference Array "I" with Best Score for Kth Linear Path Match
Normalize Best Score "SC" by the Number of Input Samples in the Path
1
4

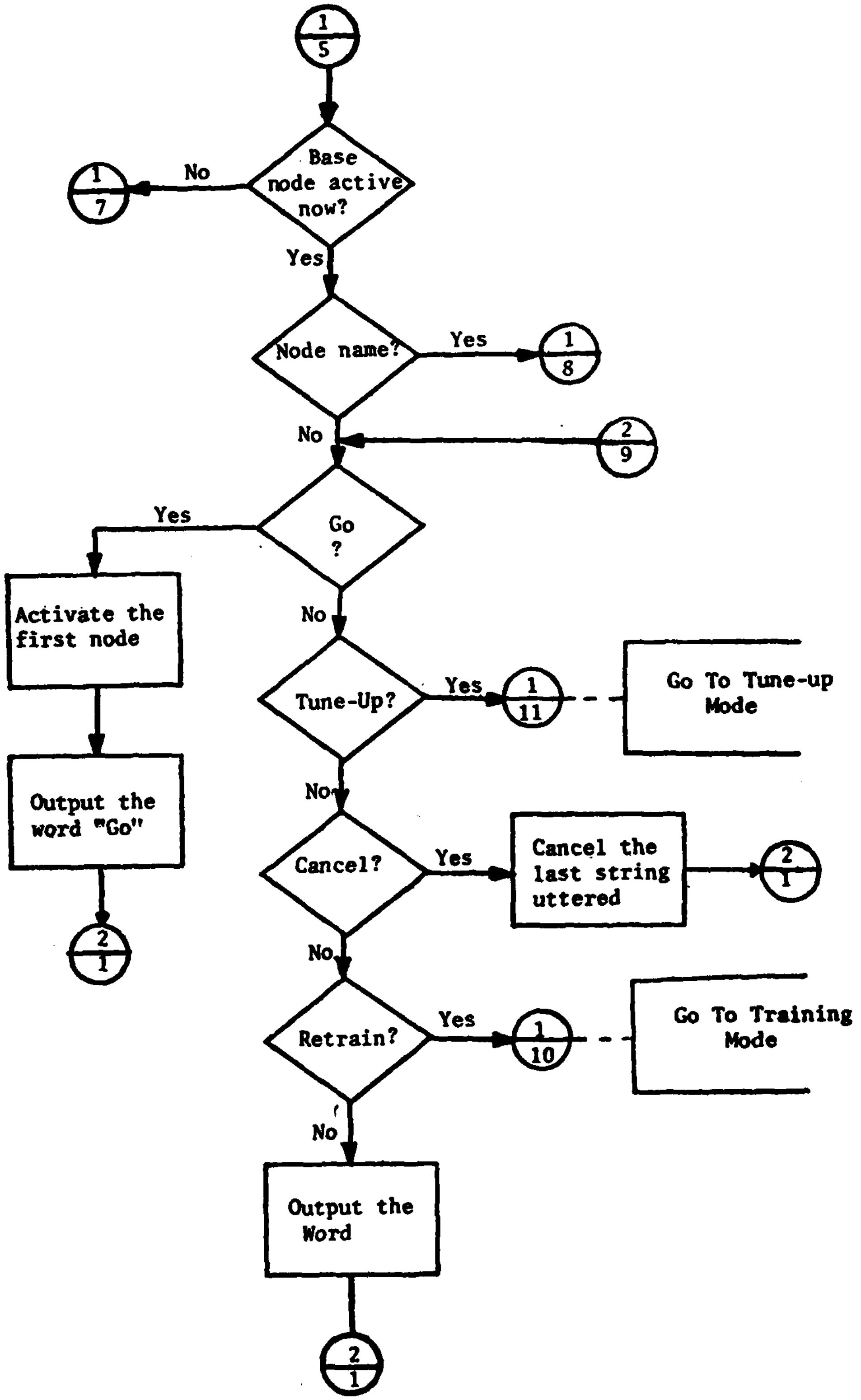
Recognition - (String Match)


1
4
Initialize "L" = 1
Obtain K^th previous best string score for L^th string
Add this score to best score "SC" just found for word "I"
Total score better than L(X)^th previous total?
Yes
Hold this total as L(X)^th best
Hold corresponding word no "I(X)"
NO
INCR "L"
L>Max #words/string ?
No
Yes
INCR "K"
K>Max #word lengths ?
No
2
3
Yes
1
5

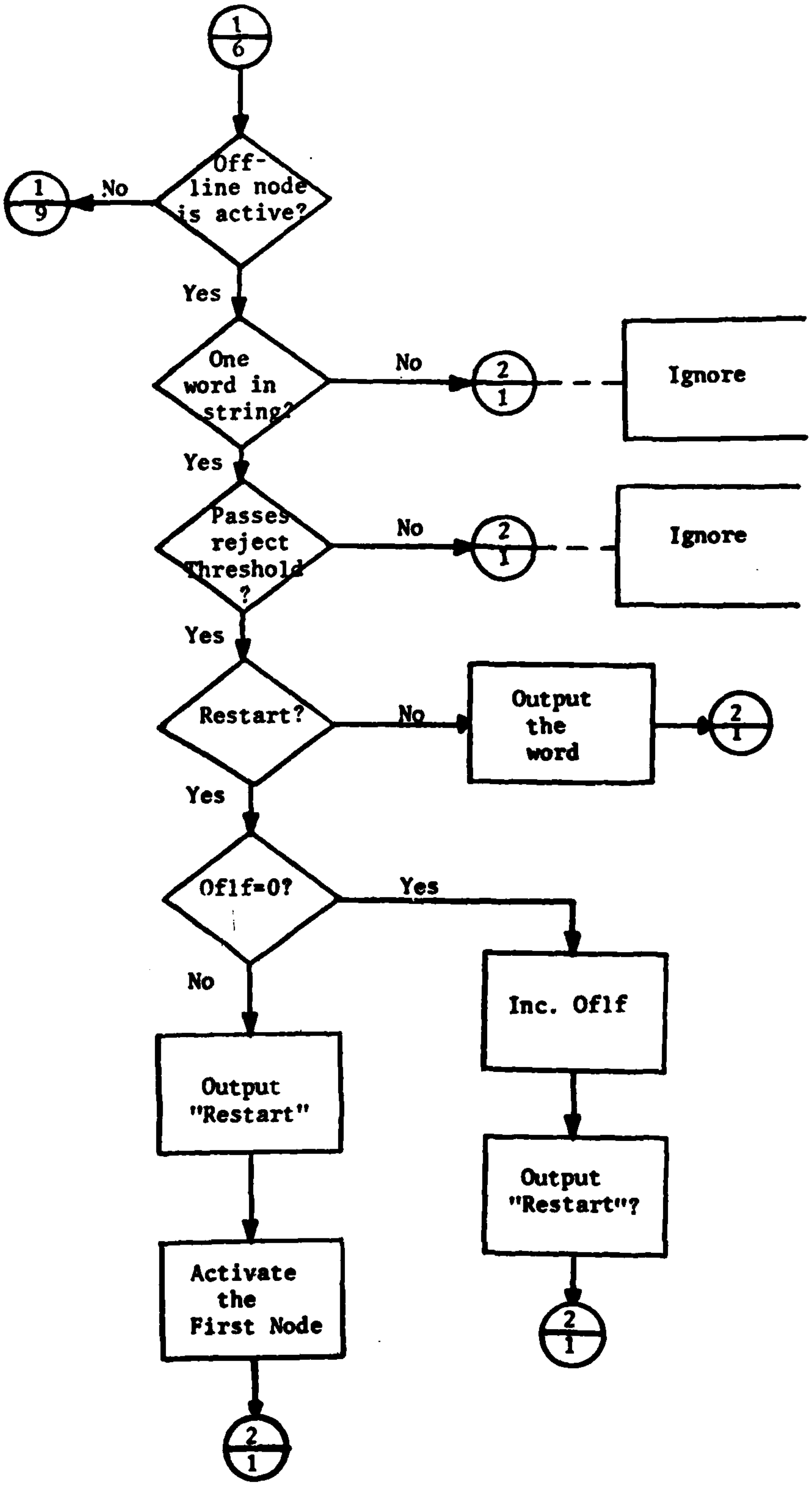
Recognition - (Loop Control)


1
4
BOSH=EOSH?
Yes
(End of String)
Choose best string "L" from Lth best total scores
No
INCR "X"
(Continue string)
Decrement through L(X)'s obtaining best word # matches for Lth string
"X" index at end of array?
Yes
Reset "X" index to beginning of word match array
No
INCR "BOSH"
CLR BOSH
CLR EOSH
BOSH at end of input buffer?
Yes
Reset "BOSH" to beginning of input buffer
1
6
No
2
1

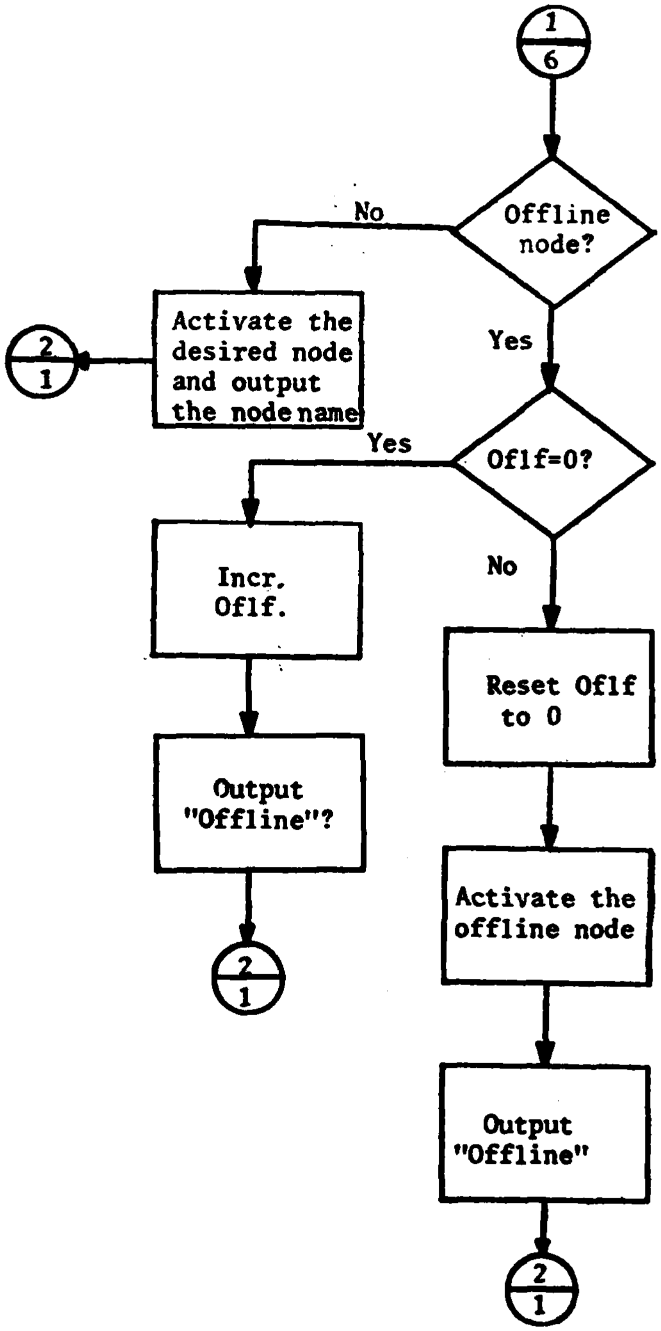
Node Structure Flow Chart



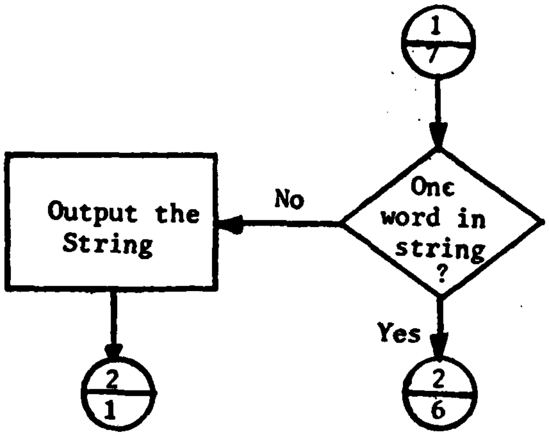
Node Structure Flow Chart



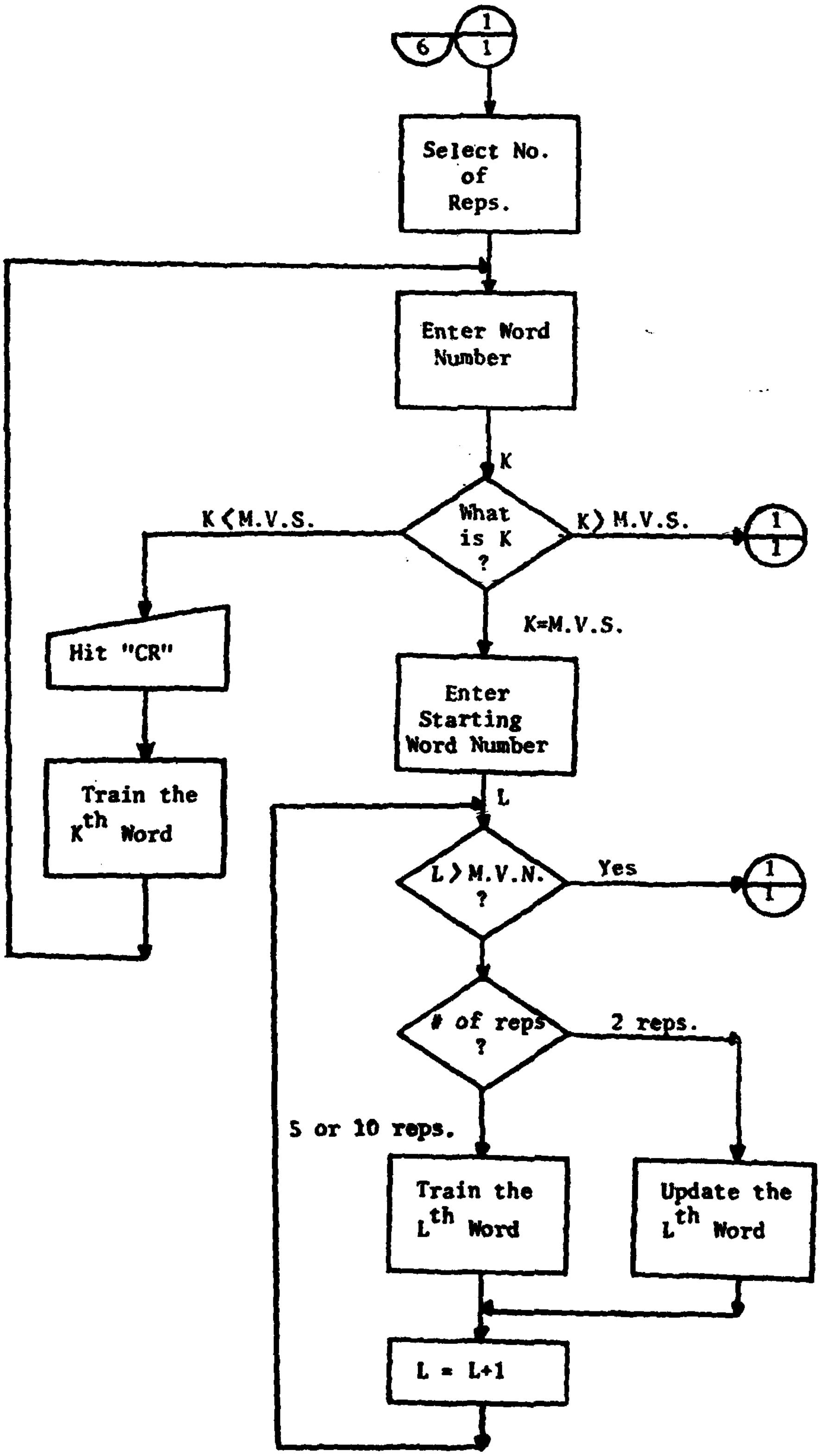
Node Structure Flow Chart



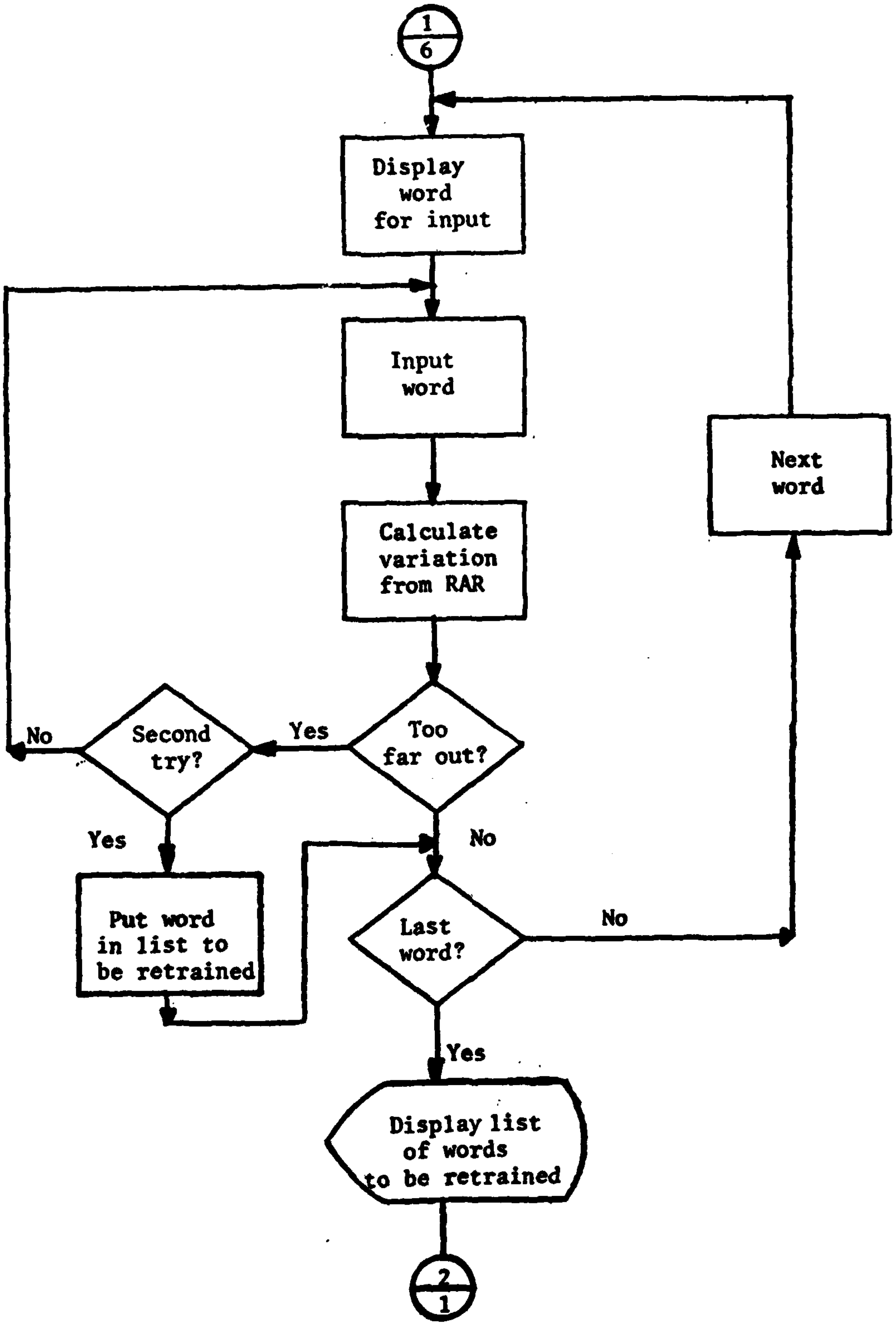
Node Structure Flow Chart



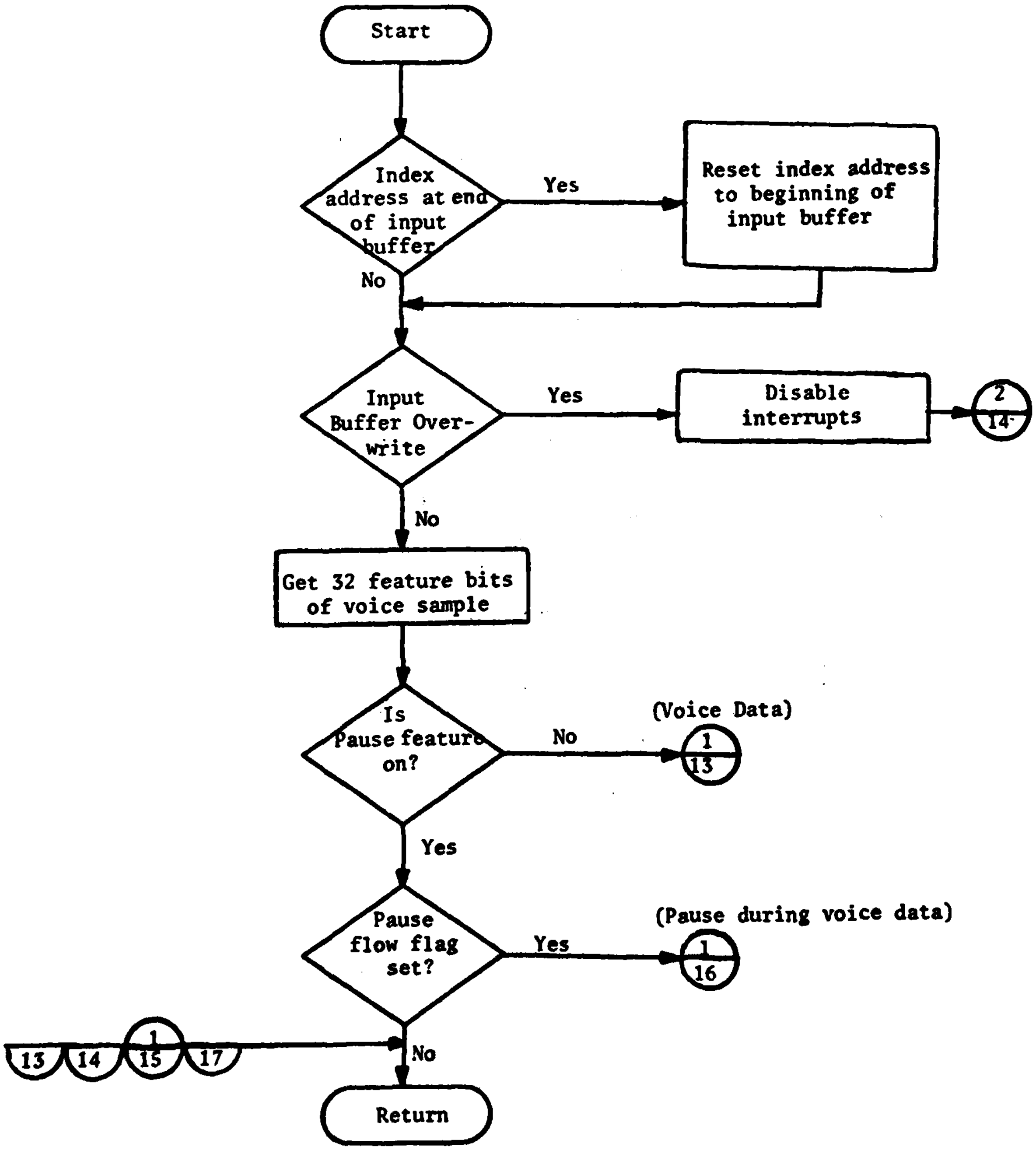
Training Routine Flow Chart



Tune-up Routine Flow Chart

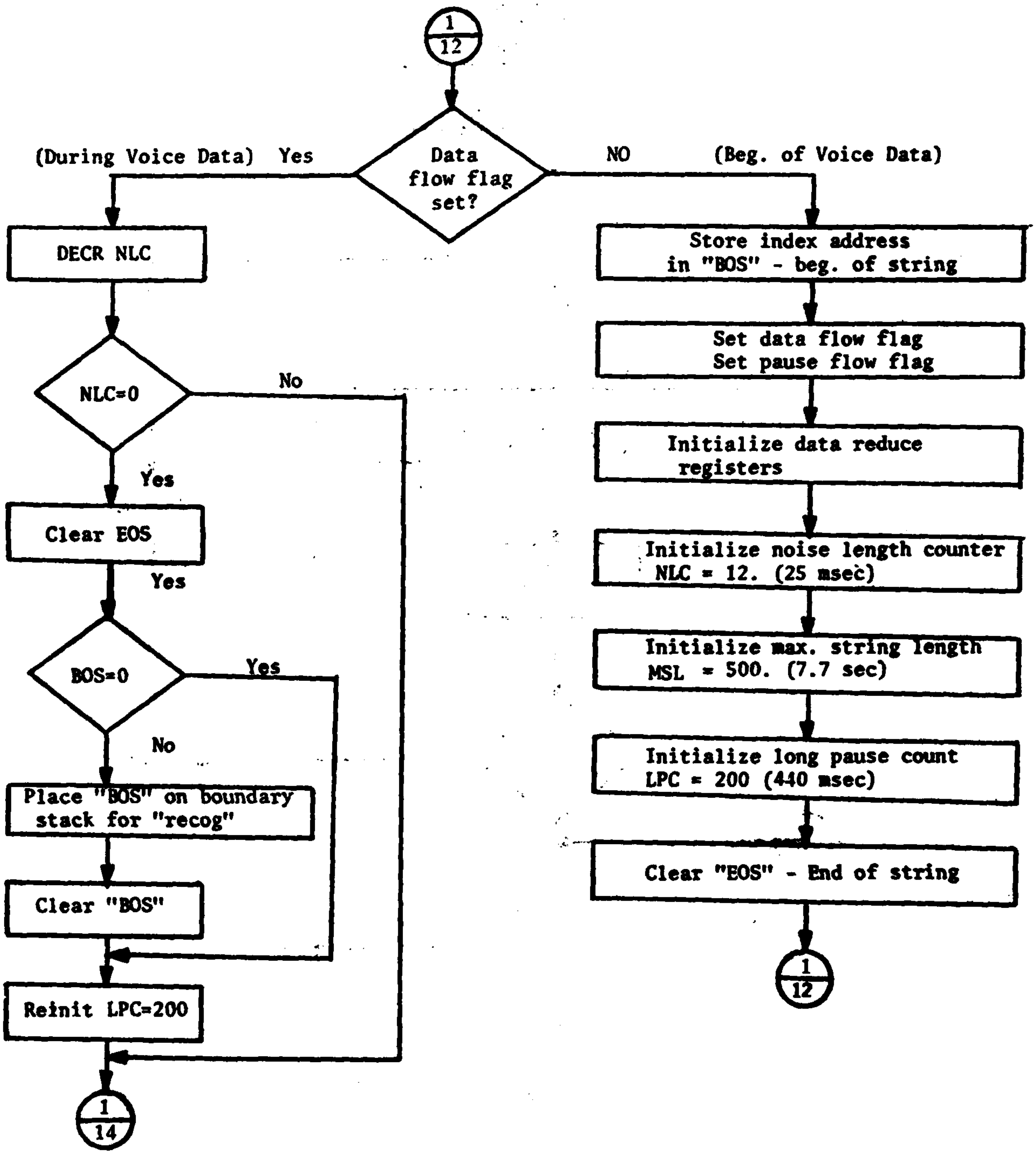

1
6
Display word for input
Input word
Calculate variation from RAR
Too far out?
Yes
Second try?
No
Yes
Put word in list to be retrained
No
Last word?
No
Next word
Yes
Display list of words to be retrained
2
1

Interrupt Service Routine Flow Chart

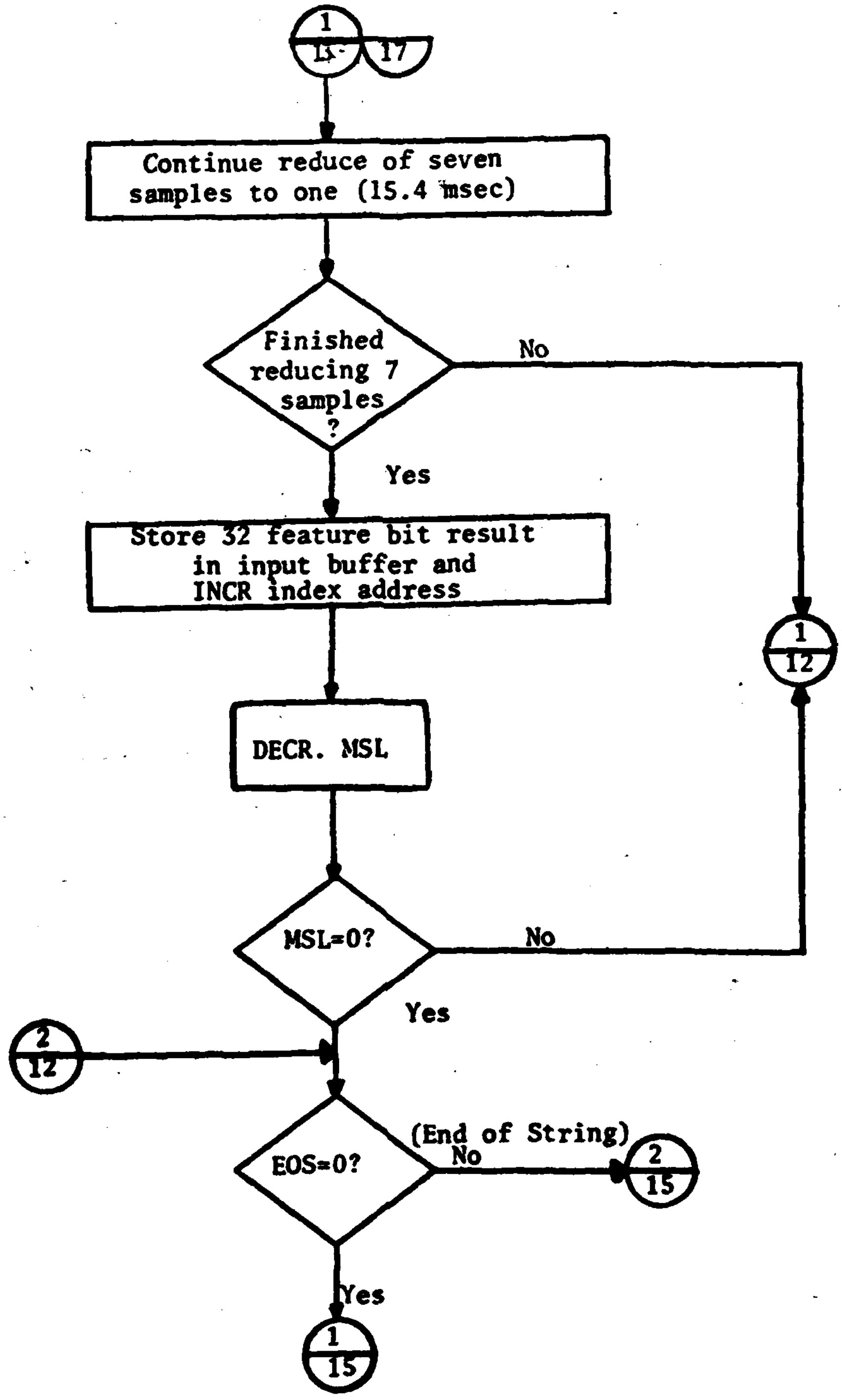

Start
Index address at end of input buffer
Yes
Reset index address to beginning of input buffer
No
Input Buffer Over-write
Yes
Disable interrupts
2
14
No
Get 32 feature bits of voice sample
Is Pause feature on?
No
(Voice Data)
1
13
Yes
Pause flow flag set?
Yes
(Pause during voice data)
1
16
13
14
1
15
17
No
Return

(Page 12 of 17)

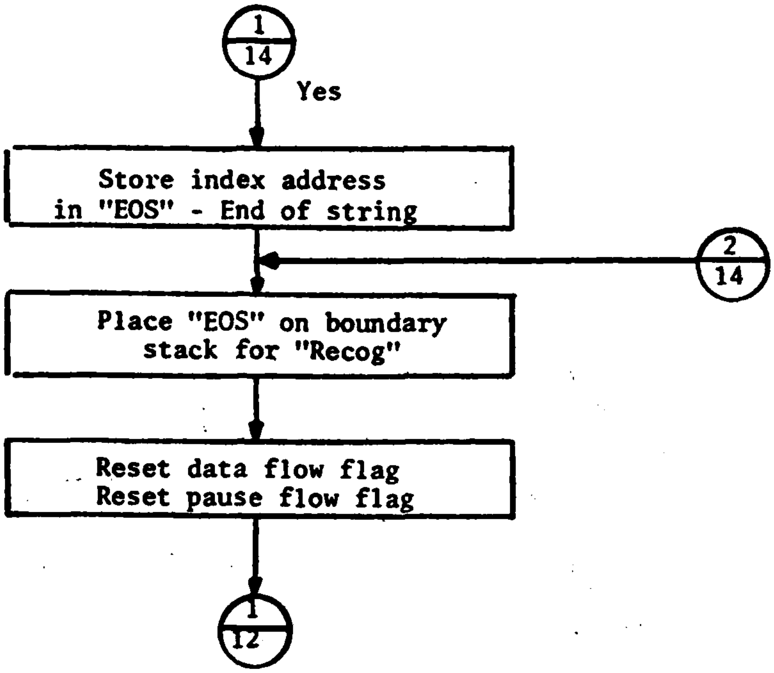
Interrupt Service Routine - (Voice Data Control)


1
12
Data flow flag set?
(During Voice Data) Yes
NO (Beg. of Voice Data)
DECR NLC
NLC=0
No
Yes
Clear EOS
Yes
BOS=0
Yes
No
Place "BOS" on boundary stack for "recog"
Clear "BOS"
Reinit LPC=200
1
14
Store index address in "BOS" - beg. of string
Set data flow flag
Set pause flow flag
Initialize data reduce registers
Initialize noise length counter NLC = 12. (25 msec)
Initialize max. string length MSL = 500. (7.7 sec)
Initialize long pause count LPC = 200 (440 msec)
Clear "EOS" - End of string
1
12

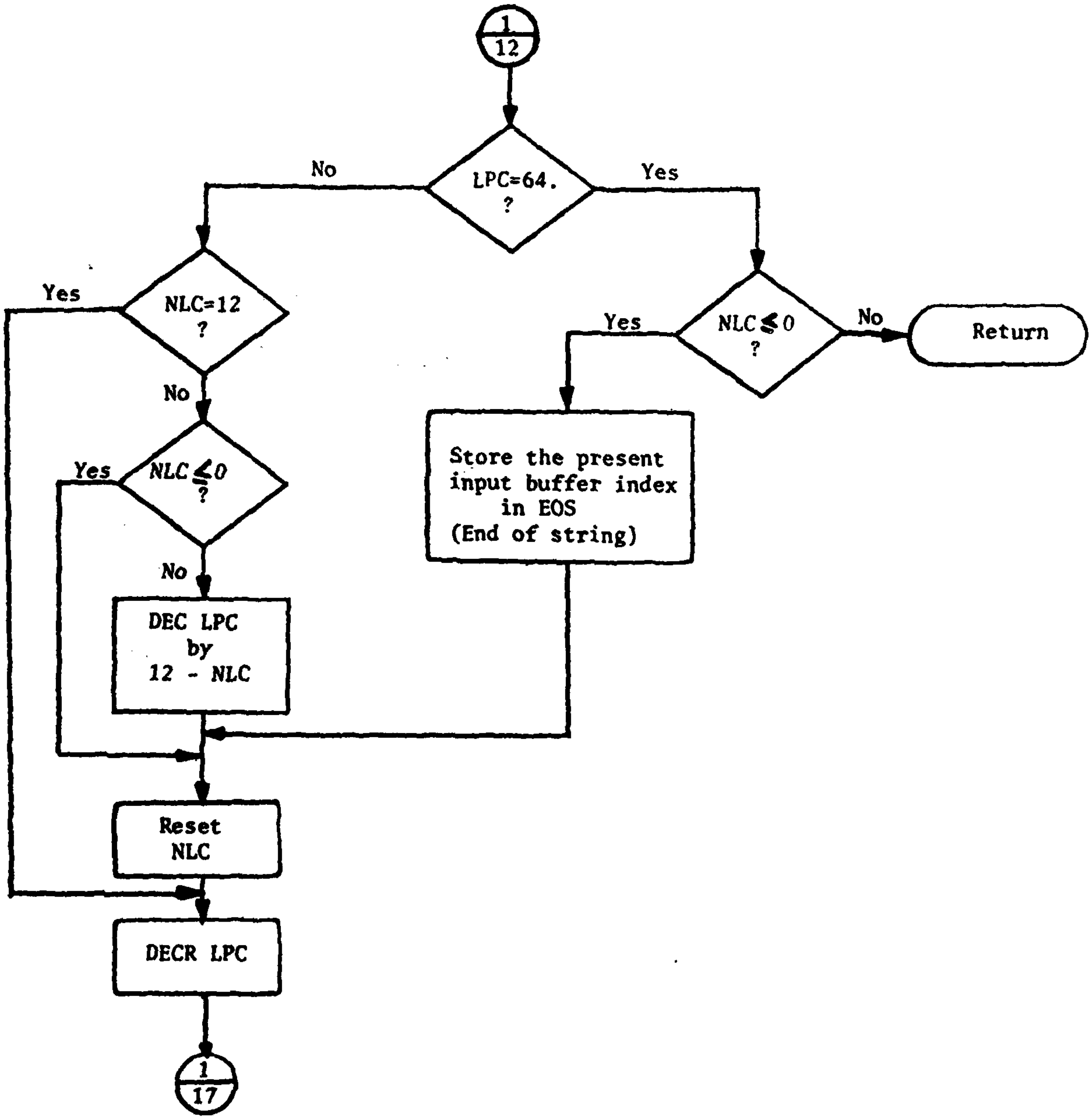
Interrupt Service Routine - (Data Reduction and Control)


1
13
17
Continue reduce of seven samples to one (15.4 msec)
Finished reducing 7 samples ?
No
Yes
Store 32 feature bit result in input buffer and INCR index address
1
12
DECR. MSL
MSL=0?
No
Yes
2
12
EOS=0?
(End of String)
No
2
15
Yes
1
15

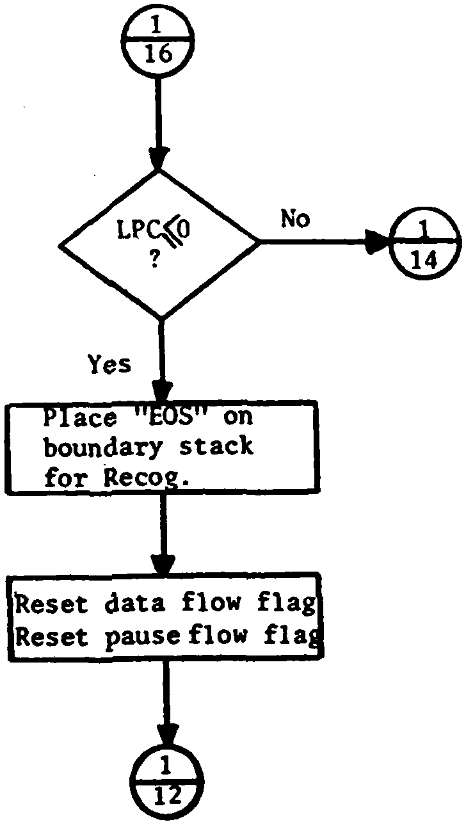
Interrupt Sērvice Routine - (Data Reduction and Control)



Interrupt Service Routine - (Pause Control)



Interrupt Service Routine - (Pause Control)



MISSION
of
Rome Air Development Center

RADC plans and executes research, development, test and selected acquisition programs in support of Command, Control Communications and Intelligence (C^3I) activities. Technical and engineering support within areas of technical competence is provided to ESD Program Offices (POs) and other ESD elements. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.

LMED

8